



## PROYECTO FIN DE CARRERA PLAN 2000

E.U.I.T. TELECOMUNICACIÓN

**TEMA:** Diseño e implementación de un instrumento virtual de medida, análisis y generación de señales.

**TÍTULO:** Implementación de instrumentación virtual con LaBVIEW

**AUTOR:** Jesús Fidel Fraile González

**TUTOR:** José Antonio Arribas Torres

**Vº Bº.**

**DEPARTAMENTO:** ELECTRÓNICA FÍSICA

**Miembros del Tribunal Calificador:**

**PRESIDENTE:** Jorge Bonache Pérez

**VOCAL:** José Antonio Arribas Torres

**VOCAL SECRETARIO:** Eduardo Nogueira Díez

**DIRECTOR:**

**Fecha de lectura:**

**Calificación:**

**El Secretario,**

### RESUMEN DEL PROYECTO:

El objetivo del PFC es el diseño e implementación de una aplicación que funcione como osciloscopio, analizador de espectro y generador de funciones virtual, todo dentro de la misma aplicación.

Mediante una tarjeta de adquisición de datos tomaremos muestras de señales del mundo real (sistema analógico) para generar datos que puedan ser manipulados por un ordenador (sistema digital).

Con esta misma tarjeta también se podrán generar señales básicas, tales como señales senoidales, cuadradas.... y además se ha añadido la funcionalidad de generar señales moduladas en frecuencia, señales tipo Chirp (usadas comúnmente tanto en aplicaciones sonar y radar, como en transmisión óptica) o PRN (ruido pseudo-aleatorio que consta de una secuencia determinista de pulsos que se repite cada periodo, usada comúnmente en receptores GPS), como también señales ampliamente conocidas como el ruido blanco Gaussiano o el ruido blanco uniforme.

La aplicación mostrará con detalle las señales adquiridas y analizará de diversas maneras esas señales. Posee la función de inventariado de los tipos de ventana mas comunes, respuesta en frecuencia, transformada de Fourier, etc.

La configuración es elegida por el usuario en un entorno amigable y de visualización atractiva.



The objective of the PFC is the design and implementation of an application that works as oscilloscope, spectrum analyzer and virtual signal generator, all within the same application. Through a data acquisition card, the user can take samples of real-world signals (analog system) to generate data that can be manipulated by a computer (digital system).

This same card can also generate basic signals, such as sine waves, square waves, sawtooth waves.... and further has added other functionalities as frequency modulated signals generation, Chirp signals type generation (commonly used in both sonar and radar applications, such as optical transmission) or PRN (pseudo-random noise sequence comprising a deterministic pulse that repeats every period, commonly used in GPS receivers). It also can generate widely known as Gaussian white noise signals or white noise uniform signals.

The application will show in detail the acquired signals and will analyze these signals in different ways selected by the user.

Windowing function has the most common window types, frequency response, Fourier transform are examples of what kind of analyzing that can be processed. The configuration is chosen by the user through friendly and attractive displays and panels.



**Escuela Universitaria de Ingeniería  
Técnica de Telecomunicación**  
Universidad Politécnica de Madrid



## Índice de contenidos

<i>Capítulo 0: Implementación de instrumentación virtual con LaVIEW</i> .....	8
<b>Introducción</b> .....	8
<b>Breve resumen: Presentación, introducción ,objetivos e historia</b> .....	8
<b>Breve resumen: Planificación</b> .....	9
<b>Breve resumen: Estado del arte</b> .....	9
<b>Breve resumen: Planificación de recursos y costes</b> .....	9
<b>Breve resumen: Ingeniería del Hardware</b> .....	10
<b>Breve resumen: Ingeniería del Software</b> .....	10
<b>Breve resumen: Integración y Test</b> .....	11
<b>Breve resumen: Recursos adquiridos para otros proyectos y posibles mejoras</b> .....	11
<b>Breve resumen: Manual de Usuario de la aplicación</b> .....	11
<b>Marco histórico de la instrumentación virtual:</b> .....	11
<i>Capítulo 1: Planificación</i> .....	14
<b>Introducción:</b> .....	14
<b>Ciclo de vida de un sistema de información:</b> .....	15
<b>Descripción del ciclo de vida según la norma ISO-12207</b> .....	15
<b>Los procesos principales del ciclo de vida</b> .....	16
<b>Clasificación de los modelos de ciclo de vida</b> .....	20
<b>El modelo de ciclo de vida en espiral</b> .....	22
<b>Ventajas del modelo en espiral</b> .....	24
<b>Dificultades del modelo en espiral</b> .....	25
<b>El Plan de Gestión del Riesgo</b> .....	27
<b>La ausencia de modelo: el modelo CODE-AND-FIX</b> .....	28
<b>El modelo por etapas (Stage-Wise) y el modelo en cascada (Waterfall)</b> .....	29
<b>Críticas al modelo en cascada</b> .....	31
<b>Flujo secuencial.</b> .....	32
<b>Resultados totales</b> .....	32
<b>Modelos de ciclo de vida basados en prototipos</b> .....	32





# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

<b>Ventajas e inconvenientes de los modelos basados en prototipos .....</b>	<b>34</b>
<b>Elección de modelo de planificación: .....</b>	<b>34</b>
<i>Capítulo 2: Estado del arte: .....</i>	<i>38</i>
<b>Introducción: .....</b>	<b>38</b>
<b>Osciloscopio: Descripción,funcionalidades y tipos .....</b>	<b>38</b>
<b>Osciloscopios analógicos .....</b>	<b>41</b>
<b>Osciloscopios digitales.....</b>	<b>44</b>
<b>Osciloscopio de señal mixta (MSO) .....</b>	<b>49</b>
<b>Analizadores de señal digital (DSA) .....</b>	<b>49</b>
<b>Osciloscopios de mano .....</b>	<b>50</b>
<b>Osciloscopios basados en ordenador .....</b>	<b>51</b>
<b>Interfaces basadas en osciloscopios .....</b>	<b>53</b>
<b>Instrumentación virtual: .....</b>	<b>55</b>
<b>El Software en la instrumentación virtual. Introducción a LabVIEW .....</b>	<b>58</b>
<b>Conclusión .....</b>	<b>67</b>
<i>Capítulo 3: Planificación de recursos y costes .....</i>	<i>68</i>
<b>Introducción: .....</b>	<b>68</b>
<b>Valoración de tareas e hitos del proyecto .....</b>	<b>68</b>
<b>Costes para la empresa .....</b>	<b>70</b>
<b>Nóminas.....</b>	<b>74</b>
<i>Capítulo 4: Fundamentos teóricos .....</i>	<i>78</i>
<b>Introducción: .....</b>	<b>78</b>
<b>Introducción a la adquisición de datos: .....</b>	<b>78</b>
<b>Fundamentos básicos de muestreo analógico:.....</b>	<b>86</b>
<b>Especificaciones dinámicas hardware, para el generador de funciones y el analizador de espectro:.....</b>	<b>96</b>
<b>Análisis de Fourier .....</b>	<b>104</b>
<i>Capítulo 5: Ingeniería del Hardware .....</i>	<i>116</i>
<b>Introducción .....</b>	<b>116</b>



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

Interfaz hardware.....	116
Consideraciones a tomar en cuenta, en relación a la precisión en las medidas ...	134
Calibración.....	134
<i>Capítulo 6: Ingeniería del Software .....</i>	<i>136</i>
Introducción .....	136
Generador de funciones.....	137
Diagrama de bloques del generador de funciones.....	139
Descripción del funcionamiento: generador de funciones.....	143
Analizador de espectro .....	143
Descripción de funcionamiento BLOQUE DUT .....	147
Descripción del bloque DSA (Dynamic Signal Analyzer).....	148
Descripción del bloque Display .....	149
Osciloscopio.....	150
<i>Capítulo 7 Integración y test.....</i>	<i>154</i>
Introducción: .....	154
Integración de hardware al sistema: Configuración de la DAQPad-6020E.....	154
Integración del software al sistema: Unión del generador de funciones, analizador de espectro y osciloscopio virtual .....	157
Unión del bloque hardware con el bloque software.....	159
Zoom de la onda que entra en la aplicación a través la DAQPad-6020E .....	162
<i>Capítulo 8: Recursos adquiridos y posibles mejoras .....</i>	<i>163</i>
Introducción: .....	163
Reutilización de Software .....	163
Recursos obtenidos con la realización del proyecto:.....	163
Conclusiones: método COCOMO .....	169
Posibles mejoras: .....	170
Introducción de hardware entre señal y sistema de medida.....	171
Pequeña introducción al aislamiento y su problemática: .....	171
Optoacopladores.....	171



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

<i>Capítulo 9 Manual de usuario .....</i>	<i>174</i>
<b>Introducción: .....</b>	<b>174</b>
<b>Componentes y controles del sistema: .....</b>	<b>175</b>
<b>Time Domain WaveForm Graft: .....</b>	<b>176</b>
<b>Escalado de tiempos y voltios por división de la gráfica de tiempos: .....</b>	<b>176</b>
<b>Frequency Domain WaveForm Graft: .....</b>	<b>177</b>
<b>Settings: .....</b>	<b>178</b>
<b>Window: .....</b>	<b>178</b>
<b>Span &amp; Resolution: .....</b>	<b>179</b>
<b>Signal Generator: .....</b>	<b>180</b>
<b>FM Signal: .....</b>	<b>181</b>
<b>Channels Selection: .....</b>	<b>182</b>
<b>Enable Physical channels: .....</b>	<b>183</b>
<b>Averaging Mode: .....</b>	<b>183</b>
<b>Weighting Mode: .....</b>	<b>184</b>
<b>Mediciones de distorsión de señal: .....</b>	<b>184</b>
<b>Códigos de error: .....</b>	<b>185</b>
<i>Bibliografía .....</i>	<i>187</i>
<b>Capítulo planificación de recursos y costes .....</b>	<b>187</b>
<b>Capítulo Ingeniería del Hardware: .....</b>	<b>187</b>
<b>Capítulo ingeniería del software .....</b>	<b>187</b>
<b>Capítulo de fundamentos teóricos .....</b>	<b>187</b>
<b>Capítulo de recursos adquiridos y mejoras .....</b>	<b>189</b>



**Escuela Universitaria de Ingeniería  
Técnica de Telecomunicación**

**Universidad Politécnica de Madrid**



# **Capítulo 0: Implementación de instrumentación virtual con LaBVIEW**

## **Introducción**

El presente proyecto de fin de carrera tiene la intención de crear un paquete de librerías de código software en LabVIEW para la creación de un osciloscopio, analizador de espectro y generador de funciones virtuales, con el objetivo de, con una tarjeta de adquisición de datos, realizar una aplicación en la que se facilite la reutilización de los recursos diseñados para test y medida de señales periódicas y no periódicas.

Se encuentra estructurado como describen los siguientes puntos, que se enumeran y se describen de la siguiente forma:

### **Breve resumen: Presentación, introducción ,objetivos e historia**

En éste presente capítulo se realizará una breve introducción sobre los objetivos iniciales que se han planteado sobre la planificación a seguir en el desarrollo del proyecto, tales como los métodos, fases de trabajo y los medios que se pretenden utilizar... realizando una breve descripción sobre el contenido de los capítulos posteriores. Posteriormente, para finalizar el capítulo se explicará de donde viene la instrumentación virtual, sus orígenes y porqué apareció en el mundo de ingeniería.



### **Breve resumen: Planificación**

La forma de planificar un proyecto de mediana o alta complejidad es esencial para llevar a cabo y con éxito un proyecto de ingeniería, por ello se ha dedicado un capítulo exclusivo a la planificación. La planificación que se seguirá se denomina planificación WaterFall. Se explicará en que se basa ese modelo de planificación y porqué se ha elegido.

Además, con la ayuda de la herramienta Mind Manager, se presentará un esquema algo más detallado sobre las tareas o hitos que se llevarán a cabo, y el tiempo que se dedicará a cada una de ellas.

### **Breve resumen: Estado del arte**

Para que el lector entienda que es la instrumentación virtual y el porqué se ha elegido realizar esta aplicación, se explicarán más detalladamente los conceptos de instrumentación virtual y osciloscopio; que son, para que se utilizan, y el estado del arte sobre ambos conceptos (entendiendo estado del arte como un resumen y organización de los resultados de la investigación reciente tratando de agregar claridad al presente trabajo en el campo de la instrumentación virtual basada en osciloscopios. Se tratará de asumir un conocimiento general del área desarrollando una perspectiva y evaluando las principales tendencias.

### **Breve resumen: Planificación de recursos y costes**

Una vez tratado el estado del arte y saber algo más sobre la instrumentación virtual,

se realizará una valoración de las tareas e hitos del proyecto, una temporización y planificación de tareas con supuestos ingenieros y becarios trabajando en éste proyecto, simulando una planificación empresarial, con un supuesto de lo que una empresa del sector de ingeniería podría invertir, tanto en tiempo como en dinero, en la realización de este proyecto. También se hará un pequeño estudio de aplicaciones sobre las que éste proyecto podría utilizarse, en referencia a aplicaciones comerciales.



### **Breve resumen: Ingeniería del Hardware**

En toda aplicación basada en la instrumentación virtual, se necesita sistema que pueda captar la señal y/o señales para poder medir o analizar su espectro. Para realizar este proyecto se la elegido la tarjeta de adquisición de datos DAQPad-6020E de National Instruments. Este capítulo se centrará en hacer una descripción detallada sobre esta interfaz hardware que se encargará de establecer el “puente” entre el mundo real y el código software de la aplicación.

En resumen, mediante una tarjeta de adquisición de datos tomaremos muestras de señales del mundo real (sistema analógico) para generar datos que puedan ser manipulados por un ordenador (sistema digital). Así podremos usar el analizador de espectro diseñado y comparar las medidas virtuales realizadas con nuestro diseño con las medidas reales usando un osciloscopio real proporcionado por el laboratorio de la universidad, por lo que, a posteriori se podrá realizar un estudio sobre esas comparaciones y comprobar que instrumentación es más rentable sobre qué tipo de aplicaciones.

### **Breve resumen: Ingeniería del Software**

Una vez claros los requerimientos del sistema, habiendo hecho un estudio del estado del arte de los componentes de la aplicación, y un análisis de viabilidad y planificación, se va a pasar a la realización de un diseño preliminar del lo que va a ser el software que va a funcionar junto a la tarjeta de adquisición.

Después del diseño preliminar, se mostrará el diseño detallado de cada uno de los módulos, mediante capturas de pantalla del código en LabVIEW, ya depurado. También se mostrarán algunas pruebas realizadas de los módulos actuando de forma independiente. En el próximo capítulo, ‘Integración y test’, los módulos se unirán, junto a la interfaz hardware, formando la aplicación completa.



### **Breve resumen: Integración y Test**

Ya una vez funcionen los módulos software por separado, se procederá a juntar los elementos del sistema, probar su funcionamiento y de ahí se añadirá el elemento hardware de la aplicación, la tarjeta DAQ-Pad 6020E, y se realizarán pruebas para comprobar su correcto funcionamiento.

### **Breve resumen: Recursos adquiridos para otros proyectos y posibles mejoras**

El proceso de industrialización del software, exige que los ingenieros y técnicos planteen nuevas alternativas para incrementar la productividad de los desarrolladores y analistas en el desarrollo de sistemas de software.

En este contexto, hay estudios que evalúan la productividad de un desarrollador realizando una comparación entre un sistema desarrollado reutilizando software, y otro sin reutilización de software. Para ello, en la evaluación se utiliza el Método Aproximativo de Costo y Productividad (COCOMO).

Dado que en la comercialización de un proyecto, el mismo está en constante desarrollo y evolución, se proponen posibles mejoras para el futuro.

### **Breve resumen: Manual de Usuario de la aplicación**

Este capítulo introduce al usuario en el manejo de la aplicación.

### **Marco histórico de la instrumentación virtual:**





## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

La instrumentación virtual es un concepto introducido por la compañía National Instruments. En el año de 1983, **Truchard y Kodosky, de National Instruments**, decidieron enfrentar el **problema de crear un software que permitiera utilizar la computadora personal (PC) como un instrumento para realizar mediciones**. Tres años fueron necesarios para crear la primera versión del software que permitió, de una manera gráfica y sencilla, diseñar un instrumento en un PC.

De esta manera surge el concepto de instrumento virtual, definido como, "un instrumento que no es real, se ejecuta en una computadora y tiene sus funciones definidas por software." (National Instruments, 2001). A este software le dieron el nombre de Laboratory Virtual Instrument Engineering Workbench, más comúnmente conocido por las siglas LabVIEW.

A partir del concepto de instrumento virtual, se define la instrumentación virtual como un sistema de medición análisis y control de señales físicas con un PC por medio de instrumentos virtuales. LabVIEW, el primer software empleado para diseñar instrumentos en PC, es un software que emplea una metodología de programación gráfica, a diferencia de los lenguajes de programación tradicionales. Su código no se realiza mediante secuencias de texto, sino en forma gráfica, similar a un diagrama de flujo. LabVIEW es un lenguaje de programación gráfica, que se ejecuta a velocidades comparables con programas compilados en C; igualmente un instrumento virtual es un módulo de software, realizado gráficamente para que parezca un instrumento físico; tiene un panel frontal que sirve como interface interactiva para entradas y salidas, un diagrama de bloques que determina la funcionalidad del programa.

Resaltan estos autores, como característica muy importante del LabVIEW que, por ser conceptualmente simple, los estudiantes se pueden concentrar en el contenido básico del experimento, no perdiendo tiempo en actividades menos importantes, como la recolección de datos. Un instrumento tradicional, se caracteriza por realizar una o varias funciones específicas que no pueden ser modificadas. Un instrumento virtual es una combinación de elementos de hardware y software usados en un PC, que cumple las mismas funciones que un instrumento tradicional. A diferencia de un instrumento convencional, un instrumento virtual es altamente flexible y puede ser diseñado por el usuario de acuerdo con sus necesidades y sus funciones pueden ser cambiadas a voluntad modificando el programa. Estas características de los instrumentos virtuales los convierten en una herramienta didáctica muy importante para aplicarse en el aprendizaje de los estudiantes de las ciencias naturales y de ingeniería.

Más adelante se realizará una explicación detallada sobre el concepto de instrumentación virtual.



**Escuela Universitaria de Ingeniería  
Técnica de Telecomunicación**

**Universidad Politécnica de Madrid**



## **Capítulo 1: Planificación**

### **Introducción:**

La aplicación contará con los siguientes componentes:

Osciloscopio: es un instrumento de medición para la representación gráfica de señales que pueden variar en el tiempo. Es muy usado en electrónica de señal, frecuentemente junto a un analizador de espectro.

El analizador de espectro: permite visualizar en pantalla las componentes espectrales en un espectro de frecuencias de las señales presentes en la entrada.

El generador de funciones: genera patrones de señales periódicas o no periódicas tanto analógicas como digitales. Se emplea normalmente en el diseño, test y reparación de dispositivos electrónicos.

-El diseño del proyecto se realizará siguiendo la planificación Waterfall, ampliamente usada por proyectos gubernamentales y por las grandes empresas. Este modelo de planificación (explicado posteriormente más en detalle) se centra especialmente en esquematizar y planificar el diseño antes de desarrollar el software, de manera que es mucho más sencillo capturar posibles errores que se podrían cometer siguiendo una metodología de codificación y depuración sin planificación.

La razón por la cual se ha tomado la decisión de seguir este tipo de planificación, es por la posibilidad de tener un acercamiento a la realización de proyectos complejos de la vida real, además de que una correcta planificación es indispensable en cualquier proyecto de ingeniería, ya que a largo plazo supone el ahorro de muchos problemas.

Para entender bien la importancia de una buena planificación en un proyecto, primero se debe explicar que es el ciclo de vida de un sistema de información:



### **Ciclo de vida de un sistema de información:**

Según el estándar ISO-12207 el ciclo de vida de un sistema de información es el marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso.

El ciclo de vida es el conjunto de fases (o etapas) por las que pasa el sistema desde que se concibe hasta que se retira del servicio. Es decir, se trata de la estructura del proceso de producción del sistema de información. El Modelo de Ciclo de Vida indica cuáles son las actividades a realizar y el orden en que se van a realizar.

Todo ciclo de vida debe cubrir tres objetivos básicos:

1. Definir las actividades a realizar y en qué orden.
2. Asegurar la consistencia con el resto de los sistemas de información de la organización.
3. Proporcionar puntos de control para la gestión del proyecto (calendario y presupuesto). No hay que confundir este concepto con el de método o metodología, la metodología indica cómo avanzar en la construcción del sistema esto es con qué técnicas, puede determinar los recursos a utilizar o las personas implicadas en cada actividad entre otras características.

El ciclo de vida nos indica las actividades a realizar, y en qué orden, para construir un Sistema de Información. Una metodología indica cómo avanzar en la construcción del sistema, es decir, las técnicas a seguir.

### **Descripción del ciclo de vida según la norma ISO-12207**

Según la Norma ISO 12207-1, las actividades a realizar durante el ciclo de vida del software se agrupan en cinco procesos principales, ocho procesos de soporte y cuatro procesos de la organización, así como un proceso especial que permite adaptar el ciclo de vida a cada proyecto concreto. A destacar que la norma no recomienda ningún modelo concreto de ciclo de vida, ni de gestión del software, ni detalla cómo realizar ninguna de las actividades.



### **Los procesos principales del ciclo de vida**

Son aquellos que resultan útiles a las personas que inician o realizan el desarrollo, la explotación o el mantenimiento del software a lo largo del ciclo de vida. Estas personas son los compradores, los proveedores, el personal de desarrollo, los usuarios y el personal encargado del mantenimiento del software.

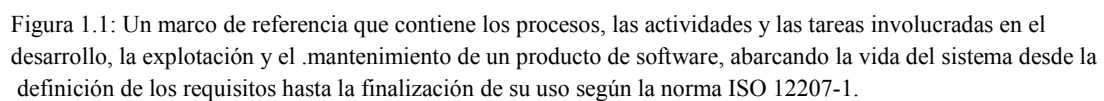
- **Proceso de adquisición:** Contiene las actividades y tareas que el comprador, el cliente o el usuario realizan para adquirir un sistema o un producto software. Aquí están incluidos la preparación y publicación de una solicitud de ofertas, la selección del proveedor del software y la correspondiente gestión de los procesos desde la adquisición hasta la aceptación del producto.
- **Proceso de suministro:** Contiene las actividades y tareas que el suministrador o proveedor realiza. Comienzan con la preparación de una propuesta para responder a una petición de oferta de un comprador o con la firma de un contrato con el comprador para proporcionarle un producto software. Trata, asimismo, de la identificación de los procedimientos y de los recursos necesarios para gestionar y garantizar el éxito del proyecto, incluyendo el desarrollo de los planes del proyecto y la ejecución de dichos planes hasta la entrega del producto software al comprador.
- **Proceso de desarrollo:** Contiene las actividades de análisis de requisitos, diseño, codificación, integración, pruebas e instalación y aceptación:
  - **Análisis de requisitos del sistema:** Aquí son especificados todos los requisitos del Sistema de Información, funciones y capacidades que debe cumplir, requisitos de seguridad, interfaces, de mantenimiento, etc.
  - **Diseño de la arquitectura del sistema:** Se identifican los principales componentes hardware y software.
  - **Análisis de los requisitos de software:** Se establecen dichos requisitos, incluyendo el nivel de calidad que debe cumplir el sistema.
  - **Diseño de la arquitectura del software:** El diseñador debe transformar el análisis anterior en una arquitectura en la que se puedan identificar sus componentes principales.
  - **Diseño detallado del software:** Aquí se realiza un diseño detallado de cada componente software, de las bases de datos y manuales de usuario.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

- Codificación y pruebas unitarias: Se desarrollan y se documentan los componentes del punto anterior.
  - Pruebas de integración: Se integran los componentes del software realizando las correspondientes pruebas.
  - Prueba del software: Las pruebas se planifican y diseñan de forma sistemática para poder detectar el máximo número y variedad de defectos con el mínimo consumo de tiempo y esfuerzo.
  - Integración del sistema: Aquí se realizan las pruebas conjuntas de los elementos hardware y software.
  - Implantación del software desarrollado en el entorno de explotación final. Cuando se sustituya a un software ya existente, puede ser recomendable un período de tiempo en el que convivan los dos sistemas.
- 
- Proceso de aceptación del software.
  
  - Proceso de explotación: Comprende la propia explotación del software y el soporte operativo a los usuarios del sistema.
  
  - Proceso de mantenimiento: Aparece cuando, tarde o temprano, el software requiere modificaciones, bien por errores, necesidades de mejora, etc.



- 18



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

- Proceso de verificación: El objetivo es demostrar la consistencia, completitud y corrección del software entre las fases del ciclo de desarrollo de un proyecto (por ejemplo, si el código es coherente con el diseño). Este proceso puede ser responsabilidad de una empresa de servicios y, en este caso se conoce como proceso de verificación independiente.
  - Proceso de validación: El objetivo es determinar la corrección del producto final respecto a las necesidades del usuario. Al igual que el anterior, este proceso puede ser ejecutado por una organización de servicios, denominándose proceso de validación independiente.
  - Proceso de revisión conjunta: Para evaluar el estado del software y sus productos en una determinada actividad del ciclo de vida o una fase de un proyecto. Las revisiones conjuntas se celebran tanto a nivel de gestión como a nivel técnico del proyecto a lo largo de todo su ciclo de vida. Un mecanismo habitual de revisión son las reuniones y la responsabilidad es generalmente compartida entre un grupo de personas pertenecientes a la organización.
  - Proceso de auditoría: Permite determinar, en los hitos preestablecidos, si se han cumplido los requisitos, los planes y, en suma, el contrato.
  - Proceso de resolución de problemas: Permite analizar y solucionar los problemas, sean éstos diferencias con los requisitos o con el contrato. Aporta un medio oportuno y documentado para asegurar que los problemas detectados son analizados y solucionados.
- Procesos de la organización: Son los utilizados por una organización para llevar a cabo funciones como la gestión, formación del personal o procesos de mejora continua.
    - Proceso de gestión: Contiene las actividades y las tareas genéricas que puede emplear una organización que tenga que gestionar sus procesos. Incluye actividades como la planificación, el seguimiento y control, la revisión y evaluación.
    - Proceso de infraestructura: Establece la infraestructura necesaria para cualquier otro proceso: hardware, software, herramientas, técnicas, etcétera para el desarrollo, explotación y mantenimiento.
    - Proceso de mejora: Para mejorar los procesos del ciclo de vida del software.
    - Proceso de formación: Para mantener al personal con la adecuada formación, lo que conlleva el desarrollo del material de formación, así como la implementación del plan de formación de la organización.





## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

- Proceso de adaptación: sirve para realizar la adaptación básica de la norma ISO 12207-1 respecto a los proyectos software. Como es sabido, las variaciones en las políticas y procedimientos de la organización, los métodos y estrategias de adquisición, el tamaño y complejidad de los proyectos, los requisitos del sistema y los métodos de desarrollo, entre otros, influyen en la forma de adquirir, desarrollar, explotar o mantener un sistema.

Dado que los procesos se aplican durante el ciclo de vida del software, y además se utilizan de diferentes formas por las diferentes organizaciones y con distintos puntos de vista y objetivos, es preciso comprender los procesos, las organizaciones y sus relaciones bajo diferentes puntos de vista:

- Contrato: El comprador y el proveedor negocian y firman el contrato, empleando los procesos de adquisición y suministro.
- Gestión o dirección: El comprador, el proveedor, el desarrollador, el operador y el personal de mantenimiento gestionan sus respectivos procesos en el proyecto software.
- Explotación: El operador proporciona el servicio de explotación del software a los usuarios.
- Ingeniería: El desarrollador o el personal de mantenimiento llevan a cabo sus respectivas tareas de ingeniería para producir o modificar los productos de software.
- Soporte: Los grupos de soporte (el de gestión de la configuración, el de aseguramiento de la calidad, el de auditoría, etcétera) proporcionan servicios de apoyo a otros grupos en el cumplimiento de tareas únicas y específicas.

### **Clasificación de los modelos de ciclo de vida**

Existen distintos modelos de ciclo de vida o lo que es lo mismo distintas pautas a seguir en el desarrollo de los Sistemas de Información. Naturalmente si se trata de proyectos sencillos en organizaciones pequeñas no es precisa la formalización del sistema. Sin embargo en



organizaciones grandes o para proyectos complejos es imprescindible establecer un modo de hacer común. Una posible clasificación sería la que divide los modelos de ciclo de vida en:

- Modelos tradicionales: Son los de más amplia utilización. Dentro de este grupo estarían:
  - Modelo en cascada
  - Modelos basados en prototipos:
    - Modelo de construcción de prototipos.
    - Modelo de desarrollo incremental.
    - Modelo de prototipado evolutivo.
- Modelos alternativos:
  - Modelo en espiral.
  - Modelos basados en transformaciones:
    - Las que usan técnicas de cuarta generación: Suelen estar basados en herramientas de cuarta generación (lenguajes no procedimentales para consultas a BD; generadores de código, de pantallas, de informes; herramientas de manipulación de datos; facilidades gráficas de alto nivel)
    - Basados en modelos de transformación.
    - Basados en herramientas CASE que permiten, siguiendo el MCV clásico, pasar de una etapa a otra aplicando las transformaciones que dan las herramientas.

En ambos casos, la filosofía general es llegar a generar código a partir de unas especificaciones transformándolas por medio de herramientas. La diferencia entre uno y otro es el uso de unas u otras herramientas.

Aparte de estos modelos de ciclo de vida en la actualidad existen nuevas alternativas:



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

- Proceso unificado de desarrollo del software de Rational (RUP).
- Modelo basado en Desarrollo de Software Basado en Componentes (DSBC o CBSB).
- Modelo de la Programación Extrema (Extreme Programming).

En la práctica, en la construcción de un Sistema de Información no se suelen seguir los modelos en su forma pura sino que de acuerdo con las peculiaridades del sistema y de la experiencia del jefe del proyecto, se pueden adoptar aspectos de otros modelos que sean más adecuados al caso concreto. Esto es así porque no existe un modelo mejor que los demás, cada uno tiene sus ventajas e inconvenientes.

### **El modelo de ciclo de vida en espiral**

Como alternativa a los modelos de ciclo de vida tradicionales se encuentra este modelo que fue propuesto por Böehm para solventar los principales problemas que presentaban aquellos. Puede considerarse como un refinamiento del modelo evolutivo general. Este modelo se caracteriza principalmente porque incorpora un nuevo elemento: el análisis de riesgos. Las principales diferencias entre este modelo y los anteriores son:

- En este modelo hay un reconocimiento explícito de las diferentes alternativas para alcanzar los objetivos del proyecto.
- Se centra en la identificación de los riesgos asociados a cada alternativa y en la manera de resolver dichos riesgos.
- Los proyectos se dividen en ciclos (ciclos de la espiral) avanzándose en el desarrollo mediante consensos al final de cada ciclo.
- Se adapta a cualquier tipo de actividad.

Es el modelo de ciclo de vida más versátil y flexible, pero también el más complejo. Cada vuelta de la espiral (ciclo) supone una refinación en el desarrollo. Cada ciclo comprende la realización de las siguientes fases:

1. Planificación. Se definen los objetivos de la fase o ciclo, así como todos los condicionantes para el cumplimiento de los mismos (restricciones de coste, tiempo, etc.) y las posibles alternativas.
2. Análisis de riesgos. Se evalúa el riesgo de cada alternativa y se elige la apropiada.
3. Ingeniería. Se construye el producto.

- 
- Progreso a través de las iteraciones
- Análisis de Riesgos
- 3
- Eje de Pto. de entrada del Proyecto
- 2
- Planificación
- 1
- Comunicación con el Cliente
- 4
- Ingeniería
- 5
- Construcción y entrega
- 6
- Evaluación del Cliente
- Proyecto de Desarrollo de Conceptos
- Proyecto de Desarrollo de Nuevos Productos
- Proyecto de Mejora de Productos
- Proyecto de Mantenimiento de Productos

Figura 1.2 Modelo de ciclo de vida en espiral

La fase de Ingeniería puede desarrollarse siguiendo cualquiera de los modelos de ciclo de vida vistos anteriormente (se dice que el modelo en espiral incluye los anteriores modelos).

En la gráfica que representa la evolución del ciclo de vida según el modelo en espiral, la dimensión radial nos permite medir el coste de la ejecución del proyecto, mientras que la dimensión angular representa el avance en su ejecución.



## **Ventajas del modelo en espiral**

La ventaja principal del modelo en espiral es que su rango de opciones acomoda las buenas características de los demás modelos de desarrollo de software, mientras su procedimiento dirigido por el riesgo, evita muchas de sus dificultades. En situaciones apropiadas, el modelo en espiral es equivalente a uno de los modelos de proceso existentes. En otras situaciones, guía la elaboración de una mezcla adecuada de procedimientos existentes para un proyecto dado.

Las condiciones bajo las cuales el modelo en espiral llega a ser equivalente a otros modelos de desarrollo de software destacados se pueden resumir como sigue:

- Si un proyecto tiene un riesgo bajo en áreas tales como el establecimiento de una interfaz de usuario no adecuada o en el cumplimiento de requerimientos rigurosos de ejecución y si, por el contrario, tiene un alto riesgo en la predicción y control del presupuesto y del calendario de elaboración, entonces. Estas consideraciones conducen el modelo en espiral a uno equivalente al modelo en cascada.
- Si los requerimientos de productos software son muy estables (implicando un bajo riesgo de diseño y fragmentación de código debido a los cambios de requerimientos durante el desarrollo) y si la presencia de errores en el producto software constituye un alto riesgo para la operación a la que se aplica, entonces, estas consideraciones sobre el riesgo conducen al modelo en espiral a asemejarse al modelo TWO-LEG de especificación precisa y desarrollo de programas formal deductivo.
- Si un proyecto tiene un bajo riesgo en áreas tales como pérdidas en el presupuesto y predicción y control de calendario, encontrándose con problemas de integración de grandes sistemas, o enfrentarse con esclerosis en la información y si hay un alto riesgo de producir una interface de usuario equivocada o de obtener unos requerimientos de soporte a las decisiones del usuario equivocados, entonces, estas consideraciones de riesgo conducen al modelo en espiral a ser equivalente al modelo de desarrollo evolutivo.
- Si se dispone de capacidades de generación de software automatizado, entonces, el modelo en espiral acomoda bien al desarrollo de prototipado rápido o a la aplicación del modelo de transformación dependiendo de las consideraciones de riesgo implicadas.
- Si los elementos de alto riesgo de un proyecto implican una mezcla de items de riesgo citado anteriormente, entonces, el procedimiento en espiral reflejará una mezcla de los modelos de proceso anteriores. Haciendo esto, sus características de eliminación de riesgos generalmente evitarán dificultades de otros modelos.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

Otras ventajas adicionales del modelo en espiral son:

- Concentra su atención en opciones que consideran la reutilización de software existente. Los pasos implicados en la identificación y evaluación de alternativas potencian estas opciones.
- Permite preparar la evolución del ciclo de vida, crecimiento y cambios del producto software.
- Proporciona un mecanismo de incorporación de objetivos de calidad en el desarrollo de producto software. Este mecanismo se deriva del énfasis puesto en la identificación de todos los objetivos restricciones durante cada uno de los círculos de la espiral.
- Es especialmente adecuado para la temprana eliminación de errores y alternativas poco atractivas. Los pasos de análisis de riesgo, validación y aceptación de compromisos cubren estas consideraciones. Para cada una de las fuentes de actividad del proyecto y gasto de recursos, se hace la pregunta claves ¿cuánto es suficiente?; dicho de otra forma: ¿cuántos requerimientos de análisis, planificación, gestión de la configuración, garantía de calidad, pruebas, verificaciones formales, etc. debería incluir el proyecto? Utilizando el procedimiento dirigido por el riesgo uno puede ver que la respuesta no es la misma para todos los proyectos y que el nivel apropiado de esfuerzo viene determinado por el nivel de riesgo incurrido por no hacer lo suficiente.
- No implica procedimientos separados para el desarrollo del software y la mejora del mismo (o mantenimiento). Este aspecto evita la frecuente consideración residual atribuida al mantenimiento del software. También ayuda a evitar muchos de los problemas que actualmente sobrevienen cuando se abordan mejoras de alto riesgo como si fuesen esfuerzos rutinarios de mantenimiento.
- Proporciona un marco viable para integrar los desarrollos de sistemas software-hardware. El centrar la atención en la gestión del riesgo y en la eliminación de alternativas no atractivas lo antes posible y con el menor coste es, igualmente, aplicable al software y al hardware.

El modelo en espiral se adapta al diseño y programación orientada a objetos (Booch, 1991), y posiblemente con estos métodos es cuando permite obtener mejores resultados: el modelo en espiral potencia la utilización de desarrollos incrementales, y cuando sea necesario, la reelaboración de partes ya desarrollados. la abstracción, encapsulación, modularidad y jerarquización, elementos fundamentales de los métodos orientados a objeto, permiten realizar fácilmente los desarrollos incrementales y hacer menos traumáticas las reelaboraciones.

### **Dificultades del modelo en espiral**





## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

El modelo en espiral completo puede aplicarse con éxito en muchas situaciones, pero es necesario tener en cuenta algunas dificultades antes de que pueda considerarse como un modelo maduro, aplicable de forma universal. Los tres primeros retos con que se enfrenta son:

- Ajustar su aplicabilidad para el caso de software contratado, con lo que ello implica (cerrar de antemano el alcance y los requisitos funcionales que debe cubrir el software, y ajustarse a unos plazos de entrega predeterminados).
- Depender en exceso de la experiencia en la evaluación de riesgos y la necesidad de una elaboración adicional de los pasos del modelo en espiral.

Adaptar su aplicabilidad al software contratado: El modelo en espiral actualmente trabaja bien en desarrollos de software internos pero necesita un trabajo adicional para adaptarlo al mundo de los contratos de adquisición del software. Los desarrollos internos de software tienen bastante flexibilidad y libertad para acomodarse a confirmaciones etapa por etapa; para diferir confirmaciones a determinadas opciones; para establecer mini espirales con las que resolver caminos críticos; para ajustar niveles de esfuerzo, o para acomodar prácticas tales como el prototipado, el desarrollo evolutivo o diseño a coste. En el mundo de la adquisición de software es muy difícil conseguir estos grados de flexibilidad y libertad sin perder responsabilidad y control y es muy difícil definir contratos que no especifiquen por adelantado los productos a obtener.

Recientemente, se ha progresado en el establecimiento de mecanismos de contratación más flexibles pero todavía se necesitan mejoras para conseguir que los compradores se sientan completamente cómodos utilizándolos.

Dependencia de la experiencia en la evaluación de riesgos: El modelo en espiral da un papel relevante a la habilidad de los desarrolladores de software para identificar y gestionar las fuentes de riesgo del proyecto. Un buen ejemplo de esto es la especificación dirigida por el riesgo en el modelo en espiral. En ella se debe llegar a un alto nivel de detalle en los elementos de alto riesgo dejando, los de bajo riesgo, para una elaboración en etapas posteriores. Un equipo de desarrolladores inexpertos o de bajo nivel pueden producir una especificación con un patrón diferente en los niveles de detalle: una gran elaboración en cuanto al detalle para los elementos de bajo riesgo y pequeña elaboración de elementos mal comprendidos, con alto riesgo. A menos que se realice una revisión de tales especificaciones por desarrolladores expertos, este tipo de proyectos ofrecerá inicialmente la ilusión de progreso durante un período, cuando en realidad el proyecto está condenado al desastre.

Otra preocupación es que la especificación será, en exceso, dependiente de las personas. Por ejemplo, un diseño producido por un experto puede ser implementado por no expertos; en



este caso, el experto que no necesita mucha documentación detallada, debe producir suficiente documentación adicional para que los no expertos no se extravíen. Con una aproximación convencional, dirigida por la documentación, el requerimiento de llevar todos los aspectos de la especificación a un nivel uniforme de detalle elimina algunos problemas potenciales y permite una adecuada revisión de algunos aspectos por revisores inexpertos. No obstante, esto produce pérdidas de tiempo a los expertos que deben buscar los aspectos críticos en medio de un gran volumen de detalles no críticos. Por otra parte, si los elementos de alto riesgo se han presentado por medio de rimbombantes referencias a funcionalidades mal entendidas, como un nuevo concepto de sincronización o un nuevo DBMS comercial, existe un riesgo, aún mayor, de que el enfoque tradicional conduzca al fracaso.

Los esfuerzos para aplicar y refinar el modelo de espiral se han centrado en la creación de una disciplina de gestión del riesgo del software, incluyendo técnicas de identificación del riesgo, análisis de riesgo, priorización del riesgo, planteamiento de gestión del riesgo, y búsqueda de elementos de riesgo.

### **El Plan de Gestión del Riesgo.**

Incluso si una organización no está lista para adoptar el procedimiento completo de espiral, una característica técnica del mismo que puede ser fácilmente adaptada a cualquier modelo de ciclo de vida proporciona muchos de los beneficios de dicho procedimiento. Se trata del Plan de Gestión del Riesgo. Este plan, básicamente, asegura que en cada proyecto se haga una identificación temprana de sus componentes de riesgo más altos; desarrolla una estrategia para resolver los componentes de riesgo; identifica y establece una agenda para resolver nuevos componentes de riesgo a medida que afloran y, por último, muestra los progresos respecto al plan en revisiones mensuales.

El plan de gestión de riesgo y las técnicas para gestión de riesgo de Software proporcionan un fundamento para adaptar los conceptos del modelo en espiral a los procedimientos de adquisición y desarrollo de software más asentados.

Se pueden sacar las siguientes cuatro conclusiones:

1. El modelo en espiral, por su naturaleza de estar dirigido por el riesgo, es más adaptable a un amplio rango de situaciones que los procedimientos dirigidos por la documentación, tales como el modelo en cascada o los procedimientos dirigidos por el código, tales como el modelo de desarrollo evolutivo. Es particularmente aplicable a sistemas de software muy grande y complejo.





2. El modelo en espiral ha tenido éxito en su mayor aplicación realizada hasta ahora: el desarrollo del TRW- SPS. Sobre todo permite alcanzar un alto nivel de capacidades ambientales de soporte del software en un corto plazo y proporciona la flexibilidad necesaria para acomodar un rango de alternativas técnicas y objetivos de usuario muy dinámico.
3. El modelo en espiral no está, todavía, tan completamente elaborado como los modelos más establecidos. Por esta razón, el modelo en espiral puede ser aplicado por personal experto, pero necesita elaboración posterior en áreas tales como contratación, especificaciones, puntos de control, revisiones, calendarios e identificación de áreas de riesgo para ser completamente aplicable en todas las situaciones.
4. Algunas implementaciones parciales del modelo en espiral como el Plan de Gestión del Riesgo, son compatibles con la mayoría de los modelos de proceso actuales y son muy útiles para superar las mayores fuentes de riesgo de proyectos.

### **La ausencia de modelo: el modelo CODE-AND-FIX**

El modelo CODE-AND-FIX. Es el modelo básico usado en los primeros tiempos del desarrollo del software y se componía de dos pasos:

1. Escribir algún código o programa (CODE).
2. Resolver los problemas en el código (FIX).

Es un modelo poco útil pero bastante común. Además es un modelo no formal que se utiliza normalmente porque es simple, pero no porque funcione bien. Puede ser efectivo para proyectos de pequeño tamaño (literalmente para "proyectos pequeños que se intentan liquidar poco después de ser construidos", de hecho "resulta peligroso para otro tipo de proyectos que no sean pequeños").

Este no-modelo tiene tres dificultades principales:

- Después de un número de ajustes, el código se hace tan poco estructurado que los siguientes ajustes se convierten en muy costosos. Esto hizo ver la necesidad de una fase previa de diseño antes de la de codificación.
- Frecuentemente, incluso con el software bien diseñado, se ajustaba poco a las necesidades de los usuarios. Esto condujo a la necesidad de introducir una fase de análisis de requerimientos antes del diseño.



- El ajuste del código (programa) era caro debido a su poca preparación para ser validado y modificado. Este problema hizo resaltar la necesidad de la planificación y preparación de las distintas tareas en cada fase.

### **El modelo por etapas (Stage-Wise) y el modelo en cascada (Waterfall)**

Las dificultades del método CODE-AND-FIX condujeron, ya en el año 1956, al convencimiento de la necesidad de realizar el desarrollo de software en un modelo por etapas (STAGewise). En este modelo el software debe desarrollarse en etapas sucesivas (planificación, especificaciones de operación, especificaciones de codificación, codificación, prueba de cada unidad, prueba de integración, eliminación de problemas, evaluación del sistema).

El modelo en cascada (WATERFALL) es una mejora del modelo anterior y ha tenido desde los años 70 un papel fundamental en el desarrollo de proyectos software, tanto es así que se le conoce también como modelo clásico del ciclo de vida del desarrollo de software. Fue enunciado por ROYCE en 1970 y las dos mejoras sustanciales que introdujo en el modelo STAGE-WISE son las siguientes:

1. Considera la realización de bucles de realimentación entre etapas, permitiendo la resolución de problemas detectados en una etapa, en la etapa anterior. Con el fin de eliminar bucles de vuelta excesivamente largos que repercutirían en una carga excesiva de trabajo en el proceso, sólo admite vueltas a la etapa anterior.
2. Una incorporación inicial del prototipado en el ciclo de vida del software. Los prototipos se utilizan para captar especificaciones durante el análisis, o para probar posibles soluciones durante el diseño pero, generalmente, una vez cumplida su misión se deben desechar, procediéndose a una especificación formal.

El modelo en cascada ayudó a eliminar muchas dificultades previamente encontradas en proyectos de desarrollo de software. Algunas de sus dificultades iniciales han sido subsanadas en sucesivas extensiones del modelo que permite considerar desarrollos paralelos, familias de programas, acomodación de cambios evolutivos, desarrollo y verificación formal del software, validación por etapas y análisis de riesgos.

Sin embargo, incluso con profundas revisiones y mejoras, el esquema básico del modelo en cascada ha seguido presentando algunas dificultades fundamentales, lo que ha llevado a buscar modelos alternativos.



Una dificultad principal del modelo en cascada es su énfasis en documentos totalmente elaborados como criterio de terminación de las distintas fases de análisis de requerimientos y diseño. Para algunos tipos de software, como compiladores o sistemas operativos, esta forma de proceder es la más efectiva. Sin embargo no es la más adecuada para otros tipos de software como, particularmente, el que se usa en las aplicaciones interactivas y de usuario final. Esta exigencia de rellenar documentos estándares ha conducido, en muchos proyectos, a escribir detalladas especificaciones para interfaces de usuario pobremente comprendidas y al desarrollo de grandes cantidades de código no utilizable.

Según este modelo tendríamos las siguientes fases en el ciclo de vida de un proyecto:

1. **Análisis:** Consiste en la recopilación de los requisitos del software. Se debe comprender el ámbito de información del software así como la función, el rendimiento y las interfaces requeridas. Estos requisitos se deben documentar y revisar de tal manera que los entiendan tanto los usuarios como el equipo de desarrollo del software. En esta fase se desarrollará el documento de requisitos del software que consistirá en una especificación precisa y completa de lo que debe hacer el sistema.
2. **Diseño:** Consiste en descomponer y organizar el sistema en elementos componentes que puedan ser desarrollados por separado. El resultado del diseño es la colección de especificaciones de cada elemento componente. En esta fase se desarrollará el Documento del diseño del Software que será una descripción del estructura global del sistema.
3. **Codificación:** En esta fase se traduce el diseño a un lenguaje legible para el computador. También se harán las pruebas o ensayos necesarios para garantizar que dicho código funciona correctamente. La documentación de esta fase será el código fuente.
4. **Integración:** Consiste en probar el sistema completo para garantizar el funcionamiento correcto del conjunto antes de ser puesto en explotación. Aquí tendremos el sistema Software ejecutable.
5. **Mantenimiento:** Puede ocurrir que durante la explotación del sistema sea necesario realizar cambios para corregir errores que no han sido detectados en las fases anteriores o bien para introducir mejoras. Tendremos que hacer un Documento de cambios ante cualquier modificación. En todas estas fases la verificación y validación se han de tener en cuenta. La verificación consiste en comprobar que el software que se está desarrollando cumple los requisitos y la validación lo que hace es comprobar que las funciones del software son las que el usuario desea.



## Modelo Cascada



Figura 1.3: Modelo en cascada

El modelo en cascada trata de aislar cada fase de la siguiente de manera que las fases sucesivas puedan ser desarrolladas por grupos de personas distintas facilitándose así la especialización. El número de fases es irrelevante, lo que caracteriza verdaderamente a este modelo es la secuencialidad entre las fases y la necesidad de completar cada una de ellas para pasar a la siguiente. El sistema está terminado cuando se han realizado todas las fases.

El modelo en cascada ayudó a eliminar muchos de los problemas que se planteaban antes de su utilización, además ha sido la base para la normalización y la adopción de estándares. A medida que ha sido utilizado se han detectado en él debilidades e inconsistencias que se han intentado corregir con diversas modificaciones y/o extensiones al modelo.

### Críticas al modelo en cascada

Las principales críticas al modelo se centran en sus características básicas, es decir secuencialidad y utilización de los resultados de una fase para acometer la siguiente de manera que el sistema sólo se puede validar cuando está terminado.



### **Flujo secuencial.**

Los proyectos reales raramente siguen el flujo secuencias que propone el modelo. Siempre ocurren interacciones y en las últimas fases sobre todo se pueden realizar en paralelo algunas áreas como por ejemplo codificación y pruebas. Una aplicación del modelo en sentido estricto obligaría a la “congelación” de los requisitos de los usuarios, supuesto este completamente alejado de la realidad. El modelo no contempla la posibilidad de realimentación entre fases.

### **Resultados totales**

El modelo en su formulación pura no prevé revisiones o validaciones intermedias por parte del usuario, así los resultados de los trabajos sólo se ven al final de una serie de tareas y fases de tal forma que si se ha producido un error en las primeras fases este sólo se detectará al final y su corrección tendrá un costo muy elevado, puesto que será preciso rehacer todo el trabajo desde el principio. El modelo no dispone de resultados parciales que permitan validar si el sistema cumple con los requisitos desde las primeras fases, dándose el caso de sistemas perfectamente formalizados y documentados que no cumplen los requisitos del usuario.

### **Modelos de ciclo de vida basados en prototipos**

Estos modelos fueron creados para solventar las diferencias percibidas en los modelos clásicos. Permiten a los desarrolladores construir rápidamente versiones tempranas de los sistemas software que pueden evaluar los usuarios.

No resulta económico generar documentación durante las fases iterativas de la construcción del prototipo. Existen varios modelos derivados del uso de prototipos:

- **Prototipos rápidos, también llamados maquetas:** Los prototipos deben poder construirse con facilidad para evaluarlos en una temprana fase del desarrollo y además han de ser baratos. Otra de las características es que deben ser desarrollados en poco tiempo. También se denominan de usar y tirar.

Tienen como finalidad la de adquirir experiencia sin pretender emplearlos como productos. Una vez aceptado, el prototipo se desecha y se comienza el desarrollo desde cero, ya que generalmente el prototipo se ha construido tomando decisiones de implementación contrarias al buen criterio de desarrollo de software. El prototipo sirve para crear y validar la especificación, y para que el usuario tenga una idea de cómo será el software antes de que comience el desarrollo. Posteriormente se convierte en un modelo en cascada. El prototipo puede ser un simple dibujo en papel. Los objetivos del prototipo son:



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

- Reducir el riesgo de construir un producto que se aleje de las necesidades del usuario
  - Reducir el coste de desarrollo al disminuir las correcciones en etapas avanzadas del mismo.
  - Aumentar las posibilidades de éxito del producto.
- Prototipado evolutivo: Fue enunciado por James Martin. En él se construye una implementación parcial del sistema que satisface los requisitos conocidos, la cual es empleada por el usuario para comprender mejor la totalidad de requisitos que desea. Responde al enunciado “no sé qué quiero, pero cuando vea algo te lo digo”. Estos modelos se encaminan a conseguir un sistema flexible que se pueda expandir de forma que sea posible realizar rápidamente un nuevo sistema cuando los requisitos cambian. Están especialmente indicados en situaciones en la que se trabaja con lenguajes de 4ª generación (4GL) y cuando el usuario no sabe lo que quiere. En este modelo se asume que los requisitos desde el principio van a cambiar continuamente, además lo lógico es comenzar con los aspectos que mejor se comprendan puesto que solo se conocerán unos pocos requisitos y los restantes se tienen que ir descubriendo en sucesivas evoluciones del prototipo; cada versión que se desarrolle será una nueva versión de todo el sistema. Está relacionado con el concepto de RAD (Rapid Application Development – Desarrollo Rápido de Aplicaciones), que identifica los asistentes, plantillas y entornos de fácil y rápida creación de software (Access, Visual Basic, etc.). Para poder evolucionar el prototipo sin desecharlo es necesario hacer uso de la reutilización, por tanto está fuertemente relacionado con técnicas de desarrollo que contemplen reutilización (Orientación a Objetos, por ejemplo). El modelo de prototipado o desarrollo evolutivo (Evolutionary Development model) también tiene sus dificultades. Se le puede considerar como una nueva versión, utilizando lenguajes de programación de más alto nivel, del viejo modelo CODE-AND-FIX. Otro inconveniente que presenta es partir de la suposición, muchas veces no realista, de que el sistema operacional del usuario final será lo suficientemente flexible como para poder incorporar caminos de evolución futuros no planificados con anterioridad.
- Prototipado incremental: Incorpora conceptos del modelo en Cascada y del de Prototipos. El proyecto se desarrolla en capas. Partiendo de un modelo en





cascada se crea una primera versión, cuya funcionalidad se va ampliando a medida la especificación crece. La gran diferencia respecto al modelo evolutivo es que los requisitos sí se conocen en su totalidad, pero su implementación se va dosificando deliberadamente.

### **Ventajas e inconvenientes de los modelos basados en prototipos**

Las ventajas de los modelos basados en prototipos son:

- Los requisitos de los usuarios son más fáciles de determinar y la implantación del sistema será más sencilla debido a que los usuarios conocen lo que esperan.
- Los sistemas se desarrollan más rápidamente
- Este paradigma facilita la comunicación con los usuarios mejorándose dicha comunicación entre el analista y el usuario.

Adolecen de los siguientes problemas:

- Puede crear falsas expectativas en el usuario ya que puede ver el prototipo como si fuera el producto final
- Puede darse una fuerte intromisión de los usuarios finales en la integración
- Se producen inconsistencias entre el prototipo y el sistema final
- Para todo tipo de prototipado cabe decir que no es un paradigma apto para proyectos grandes y de larga duración ni para aplicaciones pequeñas (menos de un mes), siendo óptimo en aplicaciones y proyectos cuya duración esté fijada entre 3 y 5 meses.

### **Elección de modelo de planificación:**

Analizando detenidamente los pros y los contras de cada modelo de planificación, se ha optado por el modelo en cascada, o modelo Waterfall. Al ser un proyecto de mediana complejidad, podemos aislar las fases tanto como queramos y proceder a la validación por separado de cada una de las fases. Cabe destacar que la interfaz hardware utilizada es un modelo comercial de tarjeta de adquisición de datos, por lo tanto tenemos la certeza de que la validación de esa fase se llevará a cabo con éxito, ya que las pruebas de esta interfaz ya los ha realizado el fabricante. No así ocurre con la interfaz software en este proyecto, la cual requiere muchas más validaciones, pero eso no quiere decir que si una validación no se lleva a cabo con éxito, obligue a la congelación de los requisitos de los usuarios, ni a la congelación del proyecto hasta el arreglo del problema, es decir, el código software es independiente de las demás



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

fases, por lo que la no validación de un bloque no posibilita la realimentación entre fases. También es cierto que la validación del proyecto obliga a llevarla a cabo cuando se acabe la aplicación, por lo que no se consideran validaciones parciales del sistema completo, pero con este proyecto concreto, se cumpliría este hecho siguiendo cualquier otro modelo de planificación. La conclusión a la que se ha llegado es que éste modelo es el que más favorece a este proyecto concreto.

A continuación se presenta un esquema detallado de las fases, el tiempo de duración de cada una de ellas, etc, con la herramienta Mind Manager Professional.





# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

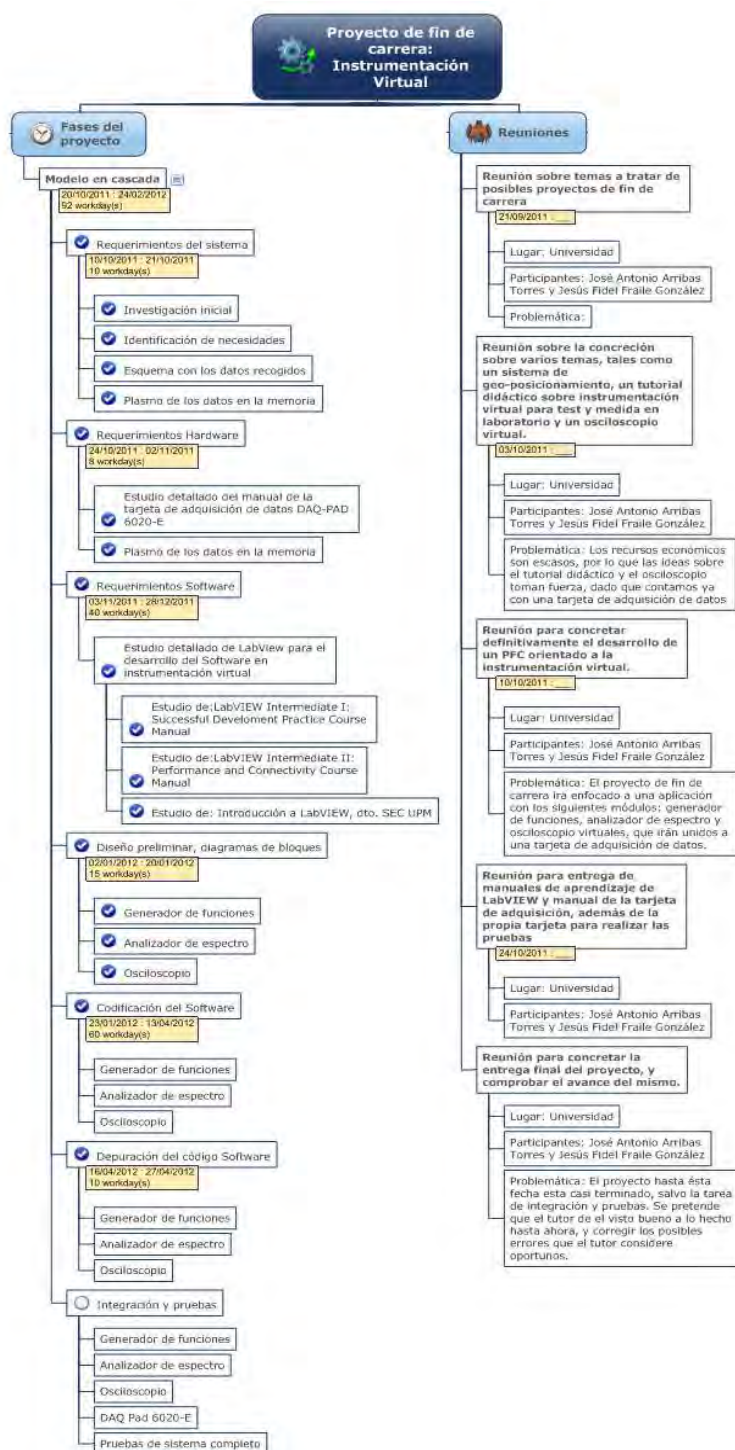


Figura 1.4 Diagrama de planificación a priori del proyecto, usando Mind Manager



**Escuela Universitaria de Ingeniería  
Técnica de Telecomunicación**

**Universidad Politécnica de Madrid**



## **Capítulo 2: Estado del arte:**

### **Introducción:**

El presente capítulo referido al estado del arte pretende resumir y organizar los resultados de investigación del campo de la instrumentación virtual y del osciloscopio, ya que ambos campos se interrelacionan para el desarrollo de la aplicación del proyecto.

Actualmente, en el mercado, se encuentran diversos tipos de osciloscopios con diferentes tecnologías, tanto para la captura, como para el tratado y la visualización de señales.

A continuación se muestran los diversos tipos de osciloscopios:

### **Osciloscopio: Descripción,funcionalidades y tipos**

El osciloscopio es una de las herramientas más importantes de la electrónica industrial, junto con el analizador de espectro de señales y el multímetro.

Un osciloscopio es un sistema de captura de señales eléctricas y electrónicas, tomadas en forma de diferencia de potencial o tensiones, con el fin de ser estudiadas en un monitor que puede estar integrado en el propio aparato o hacer uso de un monitor auxiliar mediante un PC u ordenador portátil.

Las señales analógicas muestran características como su forma de onda, su fase, su amplitud o su frecuencia.

Las señales digitales muestran características como su estado de nivel lógico, su frecuencia o su sincronismo con respecto a otras señales digitales.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

Al permitir medir las señales, estudiando su forma de onda, lo hace un valioso aliado a la hora de detectar alteraciones indeseadas: pulsos indeseados que perjudicarían a los circuitos, ruido debido a interferencias del entorno o del propio circuito, transitorios perjudiciales, etc.

En la siguiente figura podemos observar los controles de la pantalla (intensidad de la traza, enfoque y búsqueda de traza), los controles del eje horizontal y vertical y los controles de disparo.

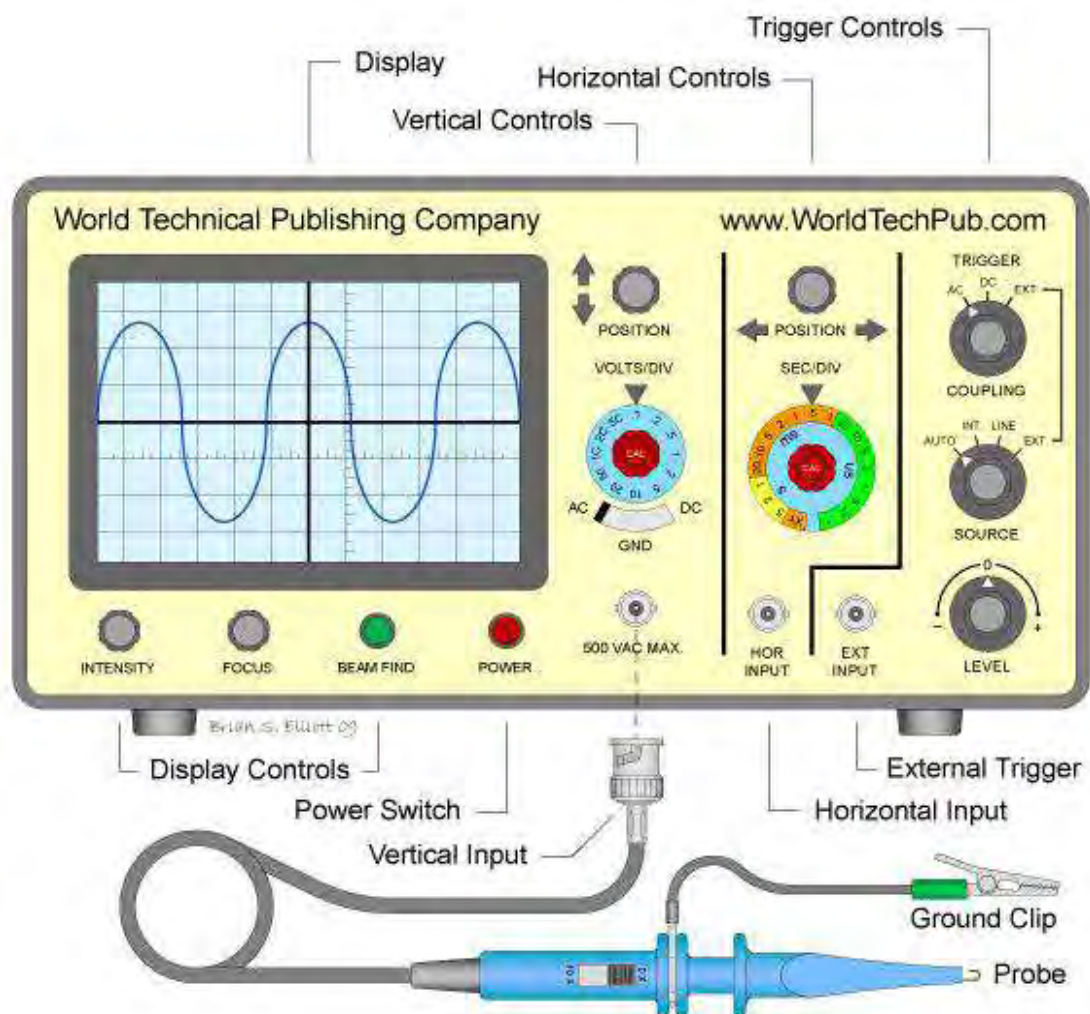


Figura 2.1: Esquema básico del panel frontal de un osciloscopio analógico.



### *Escalado de la señal*

El escalado de la señal, en los osciloscopios analógicos, se lleva a cabo con ayuda de una retícula formada por cuadrados que servirán de referencia. Los digitales normalmente numeran la escala, eliminando la retícula para una mejor lectura.

Para poder escalar correctamente la señal a estudiar, el osciloscopio dispone de una base de tiempos graduada en Time/Div, o tiempo por división, para el eje X; y de un amplificador individual para cada canal graduado en V/Div, o Volts por división, para el eje Y.

Para lograr la variación de V/Div se debe variar la ganancia del amplificador del canal representado con tal de adaptar la señal a la escala que se desee.

Para llevar a cabo el escalado de Time/Div se emplean multiplicadores y divisores de frecuencia que modifican la señal de entrada, manteniendo una proporción con respecto a la escala marcada, con tal de no falsear la medida.

### *Modo de filtrado AC o DC*

Debido a que las señales capturadas pueden disponer de componente continua y alterna al mismo tiempo, hay veces que es necesario realizar el filtrado de alguna de esas componentes.

Esto lo consigue el osciloscopio activando tres modos diferentes:

- Modo DC, en el que no se filtra ningún componente de la señal.
- Modo AC, en el que se filtra la componente continua –más concretamente las bajas frecuencias-.
- Modo GND, el cual filtra todo tipo de entrada y su uso es básicamente para referenciar la componente vertical de la señal.

### *Eje de referencia X e Y*

En ocasiones es necesario variar la posición vertical y horizontal de la señal mostrada en la pantalla, ya sea para desplazarse a través de un zoom, para variar la referencia de 0V o para visualizar mejor la fase de una señal.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

### *Función de disparo. Trigger*

El Trigger o disparador de los osciloscopios corresponde a una señal de control, ya sea por flanco de subida, de bajada o por un impulso, que indica al osciloscopio el momento en el cual debe comenzar el muestreo en la pantalla.

Esta señal puede ser controlada por el usuario de forma interna mediante un selector o bien de forma externa mediante una entrada adicional destinada a este fin.

### *Función Hold:*

Esta función se controla mediante un pulsador y permite al usuario dejar reflejada en la pantalla la señal a estudio, ignorando los futuros desencadenadores de nuevas lecturas y variaciones en la señal.

### *Ancho de banda*

El ancho de banda, BW, es una de las características más importantes de un osciloscopio y representa la gama de frecuencias que es capaz de mostrar sin atenuación. Esta atenuación es debida a los componentes que forman el osciloscopio, debido a que a frecuencias muy altas la mayoría de componentes electrónicos tienden a sufrir una caída en el valor de la tensión en sus bornes. Esto provoca que la señal medida se vea afectada y muestre valores no reales. Esta atenuación se considera aceptable siempre y cuando no supere los 3dB de ganancia, lo que corresponde a unas 0,707 veces el valor de la sensibilidad del aparato.

En aparatos analógicos viene determinado por el amplificador vertical y por el tubo de rayos catódicos. Se mediría por la capacidad de medir tiempos de subida de señales según la siguiente relación:

$$BW \text{ Hz} \cdot T_{\text{SUBIDA}} \text{ s} = 0,35$$

En aparatos digitales viene definido por la capacidad de conversión de los convertidores analógico/ digitales. Esta capacidad está relacionada por el número de muestreos por segundo y los fabricantes la dan en forma de MS/s (megamuestras/segundos). Esta capacidad debe ser diez veces superior a la frecuencia mostrada. Así una tasa de 20MS/s, por ejemplo, equivaldría a una capacidad de lectura de señales de 2MHz.

### **Osciloscopios analógicos**





Son osciloscopios, en los cuales, tanto la captura como el tratado de la señal se hace mediante señales analógicas.

Entre ellos encontramos dos tipos: osciloscopios analógicos de tiempo real y osciloscopios analógicos de muestreo.

#### *Osciloscopios analógicos de tiempo real (ART)*

Este tipo de osciloscopios realiza una captura constante en tiempo real de la señal completa y lo muestra en una pantalla de fósforo con cierta permanencia, empleando un tubo de rayos catódicos. Este tubo proyecta un haz de electrones sobre la pantalla de fósforo provocando una reacción en su superficie y volviendo luminiscente el trazado de la señal. Esta señal se muestra debidamente escalada mediante una cuadrícula de 10 de ancho por 8 de alto.

La permanencia del fósforo se puede, o bien ajustar para que dure un tiempo prolongado de varios minutos o más, o bien se puede borrar de la pantalla, todo ello mediante un circuito electrónico.

Estos osciloscopios fueron comercialmente substituidos por los osciloscopios analógicos de muestreo, no obstante permiten ver con mayor detalle señales de carácter transitorio o no repetitivas.

Su ancho de banda se encuentra alrededor de los 20MHz de frecuencia. Esto muestra un gran retraso frente a los avances tecnológicos de hoy día que emplean frecuencias del orden de GHz.

Los precios de mercado de estos aparatos se encuentran des de los 250 € en anchos de banda de 10MHz, a los 500 € en anchos de banda de 20MHz.

#### *Osciloscopios analógicos de muestreo (ASO)*

El teorema de muestreo desarrollado por Nyquist da origen al principio de muestreo en los osciloscopios. Este consiste en muestrear pequeñas porciones de la señal de entrada, llamado periodo de muestreo, en lugar de su totalidad como se hace en los osciloscopios analógicos. Posteriormente estas muestras se montan, superponiéndolas, para formar la señal completa.

Esto supone varias limitaciones:



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

- Las señales deben ser periódicas para poder tener una traza estable en la pantalla, ya que es esta periodicidad la que refresca la traza.
- Las señales extremadamente rápidas reducen el brillo de la traza dificultando su estudio.
- Las señales lentas no forman una traza debido a que la persistencia del fosforo es limitada y la retina no es capaz de retener la traza completa debido a su baja frecuencia.
- Solo se pueden ver transitorios repetitivos. Puede que el objeto de estudio sea precisamente una alteración de la señal momentánea, en forma de pulso, por ejemplo, que afecte a nuestro circuito. Mediante este tipo de osciloscopios dicha alteración se vería obviada debido a la superposición de las muestras.

Estos osciloscopios son capaces de alcanzar anchos de banda de 300MHz, aumentando así su campo de aplicación.

El precio de este tipo de osciloscopios varía desde los 550 €, con un ancho de banda de 35MHz, a los 3.000 €, con un ancho de banda de 300MHz.





Fi

gura 2.2: Osciloscopio Analógico HAMEG 2005-2 de 200MHz de ancho de banda.

## Osciloscopios digitales

A diferencia de los osciloscopios analógicos, los digitales trabajan con señales binarias correspondientes a los valores de la señal medida en cada momento. Esto se consigue transformando la señal de entrada analógica a una señal digital mediante un dispositivo llamado convertidor analógico/digital (A/D).

La entrada vertical, en lugar de ser llevada al amplificador del deflector vertical del tubo de rayos catódicos directamente, antes, es transformada por el convertidor A/D y almacenada en la memoria del microprocesador. Esto permite que la duración del almacenaje de la señal esté determinada por el tamaño de la memoria. Esto es una gran ventaja ya que nos permite ver con mayor detalle la evolución de una señal.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

Entre los osciloscopios digitales podemos encontrar tres tipos: los osciloscopios digitales de almacenamiento, los osciloscopios digitales de muestreo y los osciloscopios digitales de fósforo.

### *Osciloscopio digital de almacenamiento (DSO)*

Este es el tipo de osciloscopio más empleado en la actualidad para la mayoría de aplicaciones industriales.

Se han substituido las pantallas de rayos catódicos por pantallas LCD, las cuales permiten colores diferentes para cada canal, y un consumo menor de energía del aparato. Además de que no se necesita un segundo convertidor digital/analógico dado que este tipo de pantallas trabaja con señales digitales.

Al ser tratada la información de forma digital permite transportarla, ya sea a una red de telecomunicaciones o a un aparato de impresión o almacenaje externo al propio osciloscopio.

Los osciloscopios digitales están limitados por la circuitería analógica de captura de la señal, por la frecuencia de muestreo del mismo, la velocidad de trabajo del microprocesador y la eliminación de información aportada por la variación de intensidad de la traza.

Este tipo de osciloscopios permiten también capturar la señal de muestra en modo de tiempo equivalente. Es decir, a pesar de que cada muestra equivaldrá a la señal completa, podremos equiparar este tiempo de muestreo a la frecuencia de la señal con la ayuda de un reloj de frecuencia. Lo que obtendremos es una señal que se va actualizando, superponiéndose una muestra sobre otra -teniendo siempre en cuenta que es la señal completa- y proporcionando una definición mucho más clara de la señal que observamos. Esto nos ayudará a reducir la amplitud del ruido que interfiere en la señal muy pequeña, con respecto a la muestra que obteníamos mediante el modo de muestreo en tiempo real.

Ventajas sobre el osciloscopio analógico:

- Una pantalla más grande, con mayor brillo y con color para ver diferentes trazas.
- -Muestreo de tiempo equivalente, el cual permite una mejor definición para  $\mu V$ .



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

- Detección de pico (dado que tenemos todos los datos digitalizados solo debemos señalar cuál es el mayor).
- Pre-Trigger (disparador de almacenamiento previo al que se activa por defecto, tomando así datos de la señal anteriores a los mostrados en modo normal).
- Permite el uso de ampliaciones de múltiples señales.
- Permite mayores duraciones de almacenamiento (con memorias mayores).

Estos Osciloscopios son capaces de alcanzar los 200MHz, dependiendo de la circuitería analógica de captura.

Sus precios oscilan entre los 155€, para anchos de banda de 25MHz, a los 1.200€, para anchos de banda de 200MHz.

### *Osciloscopio digital de muestreo (TDS)*

Son osciloscopios muy similares a los osciloscopios analógicos de muestreo ya que se basan en el mismo principio. Se usan básicamente para altas frecuencias, es decir, para aquellas frecuencias que superan la frecuencia de muestreo de un osciloscopio digital de almacenamiento.

Al tomar muestras de la señal fraccionada y después superponerlas para la obtención de la señal completa, con la velocidad que facilitan las señales digitales, permite alcanzar los anchos de banda más altos del mercado.

Estos anchos de banda alcanzan los 1GHz y el precio de estos osciloscopios llegan hasta los 16.800€.



Figura 2.3: Osciloscopios digitales de Tektronix, modelo TDS2000B y TDS1000B.

### *Osciloscopio digital de fósforo (DPO)*

Es el tipo de osciloscopios más reciente del mercado digital. Fueron introducidos por la compañía Tektronix, una de las grandes referencias en el mundo de las compañías de instrumentación electrónica.

Emplea un procesamiento en paralelo de la información capturada, lo cual le permite tratar mejor los transitorios manteniendo grandes anchos de banda. Además combina este tipo de



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

tratamiento de la señal con la tecnología del fósforo digital; con una visualización muy similar a la empleada por los dispositivos analógicos, ampliando así, mediante la variación de la intensidad de la traza, la información extraída de una misma señal.

Como se realizaba en los osciloscopios de almacenamiento digital, la primera etapa de un osciloscopio de fósforo digital es el envío de la señal a un amplificador vertical precedido de un convertidor A/D. No obstante la continuación del proceso varía enormemente con respecto a los DSO que posteriormente envían la señal al microprocesador y este limita la velocidad de trabajo.

En el caso de los DPO, existen dos procesadores en paralelo para el tratado de la señal, de este modo la información que se está mostrando en pantalla, ocupando uno de los procesadores para este proceso, no interfiere con la continuación de captura y almacenamiento de la señal. De este modo la velocidad de visualización puede ser más lenta sin perder información y a la vez se amplía el campo de señal que se captura y almacena.

Este modo de captura disminuye en 1/30 por segundo el tiempo de captura, con lo que define mejor la señal real sin pérdida de información suplementaria.

El hecho de que el nombre del aparato sea osciloscopio de fósforo digital muchas veces induce al error de pensar que la pantalla emplea este componente químico para su visualización, como en el caso de los osciloscopios analógicos. El nombre se deriva de la similitud de traza entre ellos.

Los DPO emplean un sistema de visualización matricial, dividiendo el área de la pantalla en células compuestas por un conjunto de píxeles. A cada célula se le envía la información de la intensidad –y en algunos casos el color- con que deben iluminarse cada uno de los píxeles que la componen, creando un efecto de difusión y remarcando la señal principal y las alteraciones que la acompañan.

Debido a todas estas características, tanto de velocidad de muestreo, como de información visualizada, los DPO pueden alcanzar los 3,5GHz de ancho de banda con precios de hasta 33.900€.

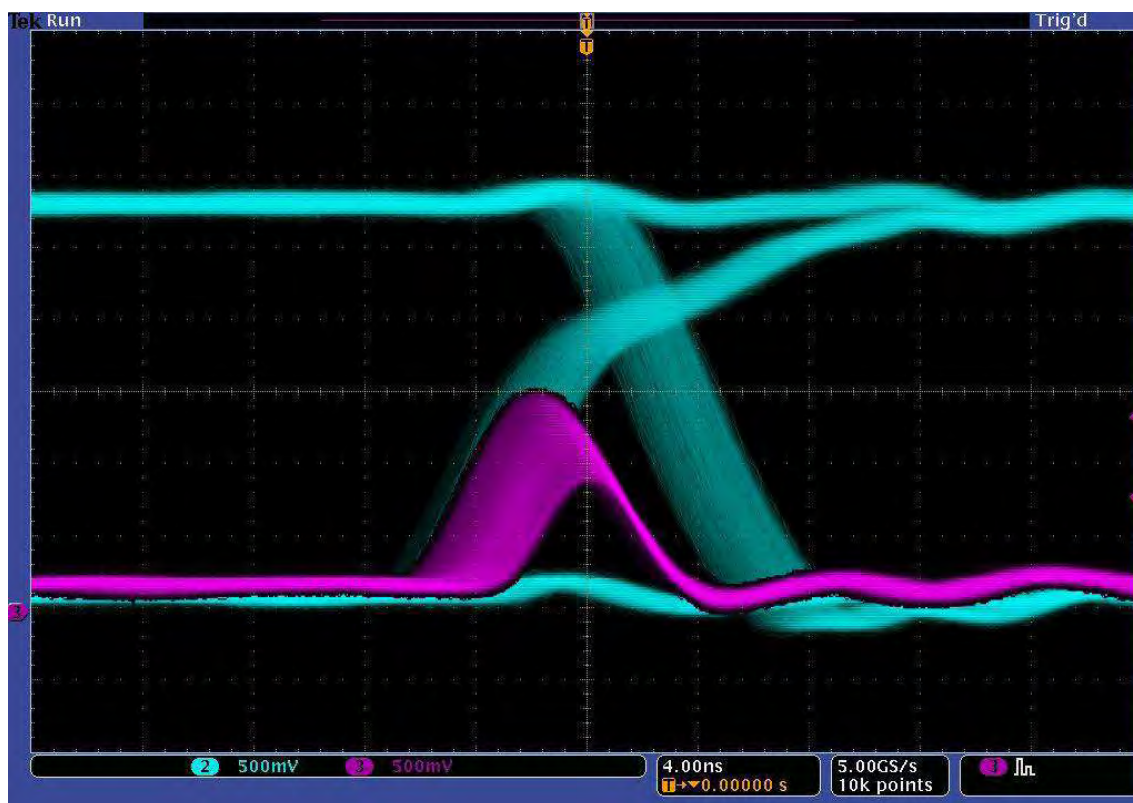


Figura 2.4: Detalle de la pantalla de un Osciloscopio DPO, modelo DPO4000 de Tektronix.

## Osciloscopio de señal mixta (MSO)

Son un tipo derivado de los osciloscopios de fósforo digital. Al ampliar los anchos de banda hasta los órdenes de GHz, se descubre la posibilidad de poder visualizar comunicaciones electrónicas en formato digital.

Esto induce a la creación de los MSO, que combinan canales analógicos y digitales según las necesidades de la medida. Esta innovación permite redistribuir el ancho de banda por los canales, en función de la aplicación que se les dé a éstos.

Los anchos de banda que emplean llegan hasta 1GHz y su precio de mercado alcanza los 17.000€.

## Analizadores de señal digital (DSA)





Según la evolución que se ha visto, el siguiente paso es el análisis de la comunicación entre puertos de un sistema electrónico o de telecomunicaciones. Para ello hablamos de anchos de banda extremadamente grandes dado a las velocidades que estos ámbitos manejan.

Otra derivación de los DPO permiten estos análisis: los DSA. Los cuales muestran por pantalla el tipo de puerto y el estado de la comunicación a través de él.

Sus anchos de banda alcanzan los 80GHz y los precios ascienden hasta los 160.000€.

### **Osciloscopios de mano**

En muchas ocasiones el usuario de estos aparatos no dispone de un lugar cómodo para realizar las medidas, o de una fuente externa de alimentación al alcance. Por estas razones se hicieron necesarios los Osciloscopios portables o de mano.

Estos aparatos disponen de baterías con autonomía limitada y de tecnologías minimizadas para poder ser transportados y empleados cómodamente.

Por norma general se emplean para estudios más básicos de las señales de los que se pueden llevar a cabo en un centro de trabajo preparado para albergar los dispositivos no portables, y es por ello que se minimizan funciones y se eliminan componentes más complejos.

Guardan en memorias internas los datos obtenidos en las mediciones para ser transportados a un centro con más prestaciones y poseen conectores USB para poder realizar el traslado de la información.

Sus anchos de banda llegan hasta los 40MHZ y su precio alcanza los 1.500€.



*Figura 2.5: Osciloscopio de mano PCE-Ibérica, modelo PCE-OC1*

## Osciloscopios basados en ordenador

Este tipo de osciloscopios tienen como objetivo el aprovechamiento de los recursos de un ordenador común para obtener un osciloscopio. Esto se consigue empleando, o bien dispositivos de captura de la señal vía conexión externa, o tarjetas integradas en el interior del propio ordenador; empleando como visualizador la propia pantalla del mismo.

Estos dispositivos buscan abaratar el uso de un osciloscopio, los cuales, como se ha visto, son de un coste en ocasiones verdaderamente elevado.

### *Captura de señal externa*

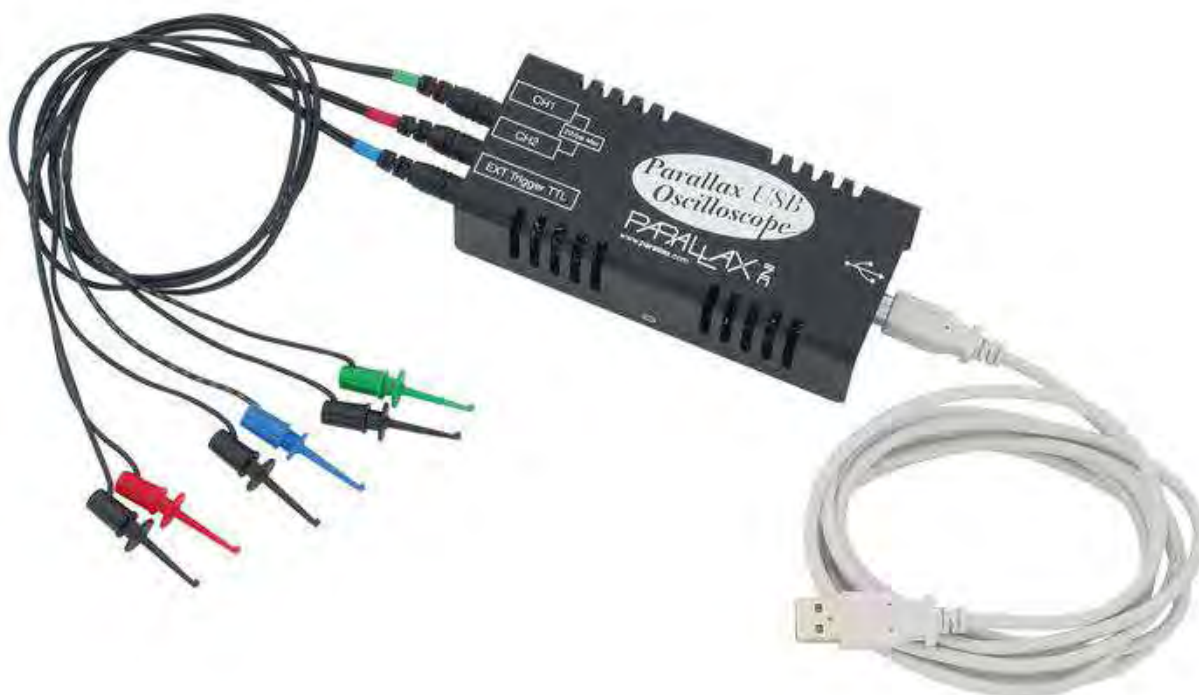
Para realizar de forma externa la captura de la señal a medir, existen hoy día diversos dispositivos periféricos, que se conectan mediante USB dado que es el modo más eficiente y rápido actualmente de enviar grandes cantidades de información empaquetada digitalmente.

Existen otros medios externos como el empleado vía puerto paralelo, puerto serie, Ethernet u otros pero no resultan tan efectivos a nivel de respuesta de escritura/lectura de los mismos.





Los anchos de banda de estos dispositivos llegan actualmente hasta los 200MHz con un coste de 220€.



*Figura 2.6: Dispositivo de adquisición de datos para osciloscopio por puerto USB de Parallax .*

### *Captura de señal por tarjeta interna*

Los dispositivos que actúan integrados dentro de una tarjeta conectada directamente al ordenador deben estar debidamente protegidos y diseñados para que no afecte al resto de circuitería propia del ordenador.



Estos dispositivos disponen de conectores mediante los cuales se accederá a los circuitos. Estos dispositivos son más raros de encontrar en el mercado y normalmente están diseñados, de acuerdo a las necesidades del usuario, por un ingeniero.

#### *Captura mediante tarjeta de audio*

Existe una forma de obtener un osciloscopio, muy limitado en ancho de banda y tensiones de entrada respecto a los anteriores. Esta consiste en la utilización de la tarjeta de sonido que ofrece, hoy día, cualquier ordenador común como tarjeta de adquisición de datos. Dichas tarjetas, al estar diseñadas para el tratamiento de señales acústicas, disponen de un ancho de banda de decenas de kHz de frecuencia correspondiente al espectro audible del ser humano. Estos dispositivos carecen de coste adicional.

#### **Interfaces basadas en osciloscopios**

Todos los sistemas de osciloscopio basados en un ordenador deben disponer de un software que actúe como interface con el usuario. Esta interface debe estar diseñada de tal forma que se identifiquen claramente los mandos habituales de un osciloscopio convencional, y por cuestiones de una familiarización más rápida, acostumbran a parecerse bastante a los osciloscopios convencionales.

Entre estas interfaces existen muy diversos tipos: desde las proporcionadas por el fabricante del los osciloscopios que efectúan la captura por puerto USB, a las muy diversas interfaces que se encuentran hoy día en internet para adaptar las entradas de línea de la tarjeta de audio del ordenador a tal fin.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

### Adquisición por dispositivos USB

Por norma general este tipo de interfaces están creadas por la misma compañía que diseña la adquisición de datos por USB. Esto provoca que haya multitud de interfaces, una para cada marca dedicada a este tipo de Osciloscopios. Se pondrá el ejemplo del software que aporta la empresa Parallax.

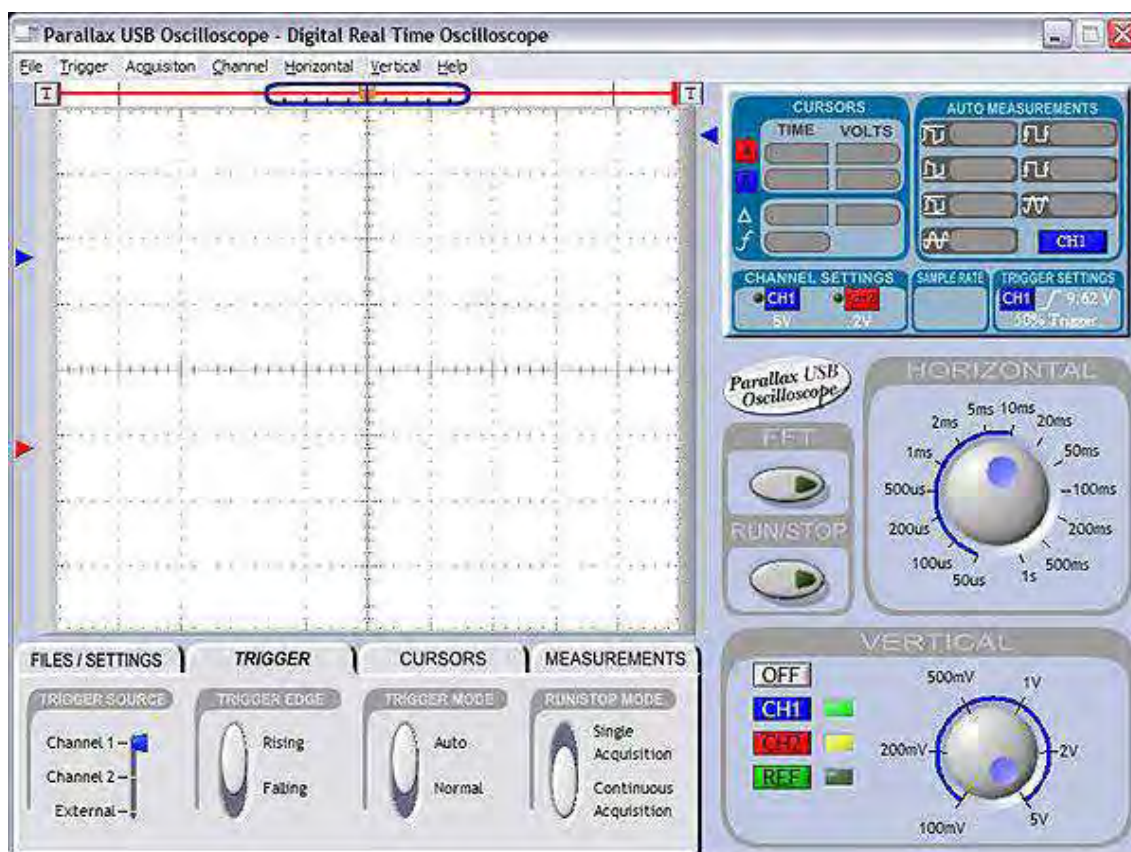


Figura 2.7: Captura de la interface correspondiente al software aportado por la compañía Parallax.

Se pueden observar la retícula de la pantalla en formato 10x8, los selectores correspondientes al escalado del eje vertical y horizontal y la función Trigger en las pestañas del menú.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

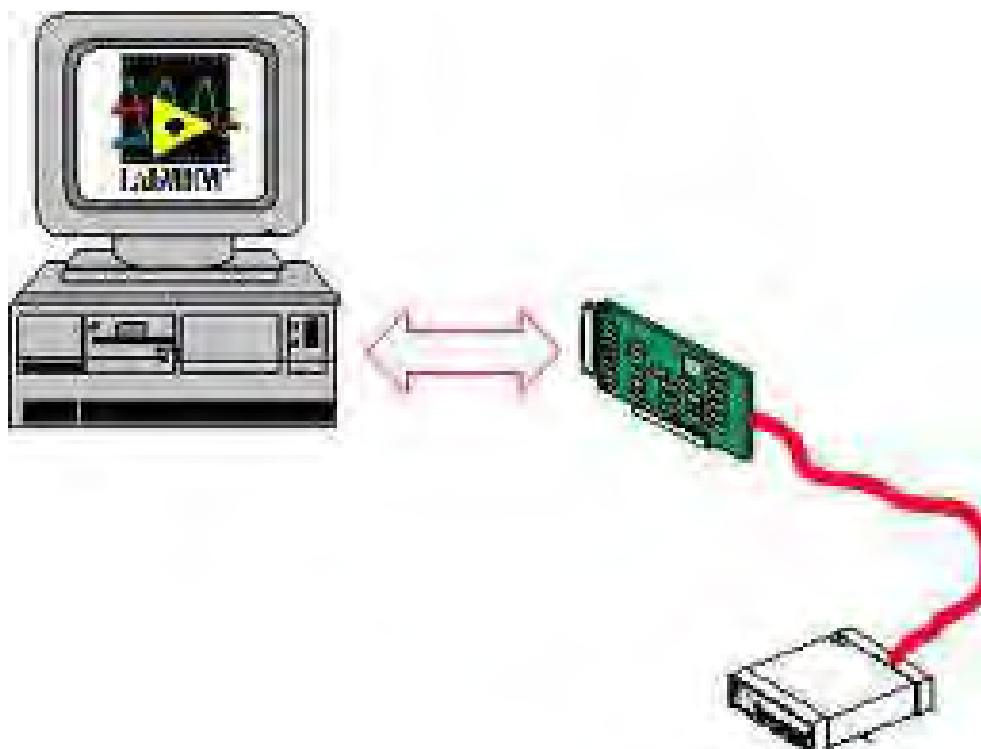
Existen otras marcas, como USBee que van más allá, incluyendo en el software un sistema de analizador de espectros y un sistema analizador lógico para buses de comunicaciones.

Este software entra dentro del precio del propio sistema de adquisición de datos.

### **Instrumentación virtual:**

La rápida adopción del PC en los últimos 30 años generó una revolución en la instrumentación de ensayos, mediciones y automatización. Un importante desarrollo resultante de la ubicuidad del PC es el concepto de instrumentación virtual, el cual ofrece muchos beneficios a ingenieros y científicos que requieran mayor productividad, precisión y rendimiento.

Un instrumento virtual consiste en un ordenador del tipo industrial, o una estación de trabajo, equipada con poderosos programas (software), hardware económico, tales como placas de inserción, y drivers que cumplen, en conjunto, las funciones de instrumentos tradicionales. Los instrumentos virtuales representan una alternativa fundamental de los sistemas de instrumentación basados en el hardware, ya que los sistemas centrados en el software aprovechan la potencia de cálculo, productividad, exhibición y capacidad de conexión de las populares computadoras de escritorio y estaciones de trabajo. Aunque el PC y la tecnología de circuitos integrados han experimentado avances significativos en las últimas tres décadas, es el software el que realmente provee la ventaja para construir sobre esta potente base de hardware para crear los instrumentos virtuales, proveyendo mejores maneras de innovar y de reducir los costos significativamente. Con los instrumentos virtuales, los ingenieros y científicos construyen sistemas de medición y automatización que se ajustan exactamente a sus necesidades (definidos por el usuario) en lugar de estar limitados por los instrumentos tradicionales de funciones fijas (definidos por el fabricante).



*Figura2.8: Esquema adquisición de datos*

Un hardware de adquisición de datos flexible y la computadora personal, son los componentes esenciales de la instrumentación virtual. La sinergia entre ellos ofrece ventajas que no pueden ser igualadas por la instrumentación tradicional.

### Instrumentos virtuales contra instrumentos tradicionales

Los instrumentos autónomos tradicionales, tales como osciloscopios y generadores de ondas, son muy poderosos, caros y diseñados para llevar a cabo una o más tareas específicas definidos por el fabricante. Sin embargo, el usuario por lo general no puede extender o personalizar esas tareas. La interfaz de entrada/salida del instrumento, sus circuitos electrónicos y las funciones disponibles para el usuario son todas específicas a la naturaleza del instrumento. Además, debe desarrollarse una tecnología especial y costosos componentes para construirlos, lo cual los hace muy caros y lentos para adaptarlos. Debido a que están basados en computadores personales, los instrumentos virtuales aprovechan inherentemente los beneficios de la última tecnología de los PCs corrientes. Estos avances en tecnología y rendimiento, que están cerrando rápidamente la brecha entre los instrumentos autónomos y los PCs, incluyen poderosos procesadores, sistemas operativos y tecnologías tales como el



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

Microsoft Windows XP, .NET, etc. Además de incorporar características poderosas, esas plataformas también ofrecen un acceso sencillo a herramientas también poderosas tales como Internet. Los instrumentos tradicionales también sufren frecuentemente de falta de portabilidad, en tanto que los instrumentos virtuales que corren en los ordenadores portátiles automáticamente incorporan esta naturaleza portátil. Los ingenieros y científicos cuyas necesidades, aplicaciones y requerimientos varían muy rápidamente, necesitan flexibilidad para crear sus propias soluciones. El usuario puede adaptar un instrumento virtual a sus necesidades particulares sin necesidad de reemplazar todo el instrumento dado que posee el software de aplicación instalado en la computadora y al amplio rango disponible de hardware para instalar en ella.

### Flexibilidad

A excepción de los componentes especializados y los circuitos hallados en los instrumentos tradicionales, la arquitectura general de los instrumentos autónomos es muy similar a la hallada en un instrumento virtual basado en computadora.

Ambos requieren uno o más microprocesadores, puertos de comunicación (por ejemplo: serie y GPIB) y capacidad de mostrar resultados así como también módulos de adquisición de datos. Lo que diferencia uno del otro es su flexibilidad y el hecho de que el usuario pueda modificar y adaptar el instrumento a sus necesidades particulares. Un instrumento tradicional podría contener un circuito integrado para llevar a cabo un conjunto particular de instrucciones de procesamiento de datos; en un instrumento virtual estas funciones podrían llevarse a cabo por el programa que corre en el procesador de la computadora. El usuario puede fácilmente extender ese conjunto de funciones y estar sólo limitado por la potencia del software que utilice.

### Reducción de Costos

Utilizando soluciones basadas en la instrumentación virtual, se pueden reducir los costos de inversión, desarrollo de sistemas y mantenimiento al mismo tiempo que mejora el tiempo de comercialización y la calidad de sus propios productos.

### Hardware para Insertar y de Red

Existe una amplia variedad disponible de hardware que se puede o bien insertar en una computadora o bien acceder a través de una red. Estos dispositivos ofrecen un amplio rango de capacidades de adquisición de datos a un costo significativamente inferior que el correspondiente a dispositivos dedicados a éste propósito. A medida que la tecnología de





## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

Los circuitos integrados avanzan y los componentes comunes se vuelven más baratos y poderosos, también lo hacen las placas que ellos utilizan. Junto con estos avances tecnológicos viene un incremento en las velocidades de adquisición de datos, precisión de las mediciones y mejor aislamiento de las señales.

Dependiendo de la aplicación en particular, el hardware que se elija podría incluir entradas o salidas analógicas, entradas o salidas digitales, contadores, temporizadores, filtros, muestreo simultáneo y capacidades de generación de ondas. La amplia gama de placas y hardware podría incluir cualquiera de estas características o una combinación de ellas.

### **El Software en la instrumentación virtual. Introducción a LabVIEW**

El software es el componente más importante de un instrumento virtual. Con la herramienta de software apropiada los ingenieros y científicos pueden crear eficientemente sus propias aplicaciones, diseñando e integrando las rutinas que requiere un proceso en particular.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

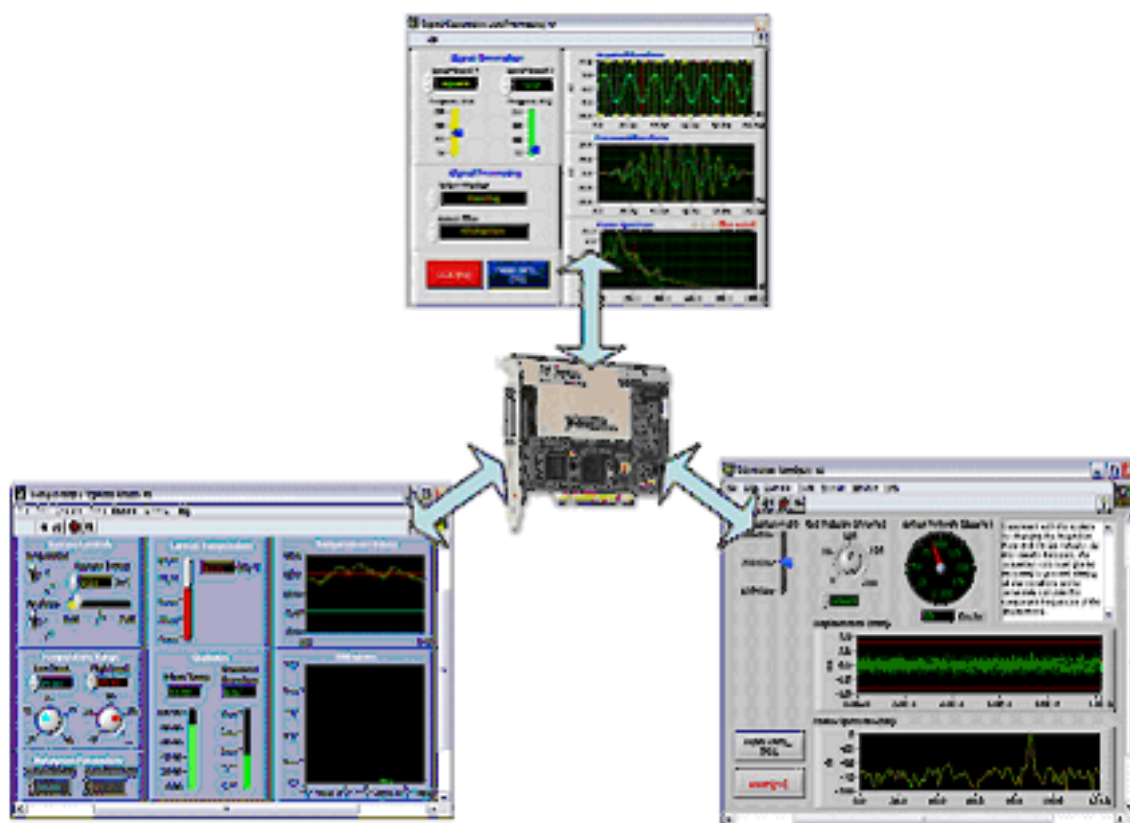


Figura2.9: Proyectos en LabVIEW

También pueden crear las interfaces de usuario que mejor satisfagan el objetivo de la aplicación y de aquéllos que van a interactuar con ellas. Pueden definir cómo y cuándo la aplicación adquiere datos desde el dispositivo, cómo los procesa, manipula y almacena los datos y cómo se presentan los resultados al usuario.

Contando con un software poderoso, el usuario puede dotar a sus instrumentos con capacidades de inteligencia y de toma de decisiones de manera tal que se adapten cuando las señales medidas varíen inadvertidamente o cuando se requiera mayor o menor potencia de procesamiento.

Una importante ventaja que provee el software es la modularidad. Cuando se trata de un gran proyecto, los ingenieros y científicos generalmente abordan la tarea dividiéndola en unidades





funcionales manejables. Estas tareas subsidiarias son más manejables y más fáciles de probar dadas las menores dependencias que podrían causar comportamientos inesperados.

El usuario puede diseñar un instrumento virtual para solucionar cada una de estas tareas subsidiarias y luego reunir las en un sistema completo para resolver la tarea de mayor envergadura. La facilidad con la cual usted puede realizar esta división de tarea depende en mayor medida de la arquitectura subyacente en el software.

## Aplicaciones Distribuidas

Un instrumento virtual no está limitado a estar confinado en una computadora autónoma. En realidad, con los recientes desarrollos en tecnologías de redes e Internet, es más común utilizar la potencia de conectividad de los instrumentos con el fin de compartir tareas.

Ejemplos típicos incluyen supercomputadoras, monitoreo distribuido y dispositivos de control, así como también datos o visualización de resultados desde múltiples sitios.

## LabVIEW como herramienta para crear instrumentos virtuales

LabVIEW es una parte integral de la instrumentación virtual dado que provee un medio ambiente de desarrollo de aplicaciones que es fácil de utilizar y está diseñado específicamente teniendo en mente las necesidades de ingenieros y científicos. LabVIEW ofrece poderosas características que facilitan la conexión a una gran variedad de hardware y otros software.

## Programación Gráfica

Una de las características más poderosas que LabVIEW ofrece a los ingenieros y científicos es un medio ambiente de programación que es gráfico. Con LabVIEW el usuario puede diseñar instrumentos virtuales a medida creando interfaces gráficas de usuario en la pantalla de ordenador con las cuales se pueden:

- Operar el programa de instrumentación
- Controlar el hardware seleccionado



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

-Analizar datos adquiridos

-Visualizar los resultados

Los usuarios podrán observar los datos del programa actualizados en tiempo real (cómo van fluyendo los datos en un panel frontal, donde se definen los controles (botones, marcadores, etcétera) e indicadores (los usamos como salidas, pueden ser gráficas, marcadores, etcétera),

a fin de emular paneles de control de instrumentos tradicionales, crear paneles de ensayo personalizados o representar visualmente el control y operación de procesos.

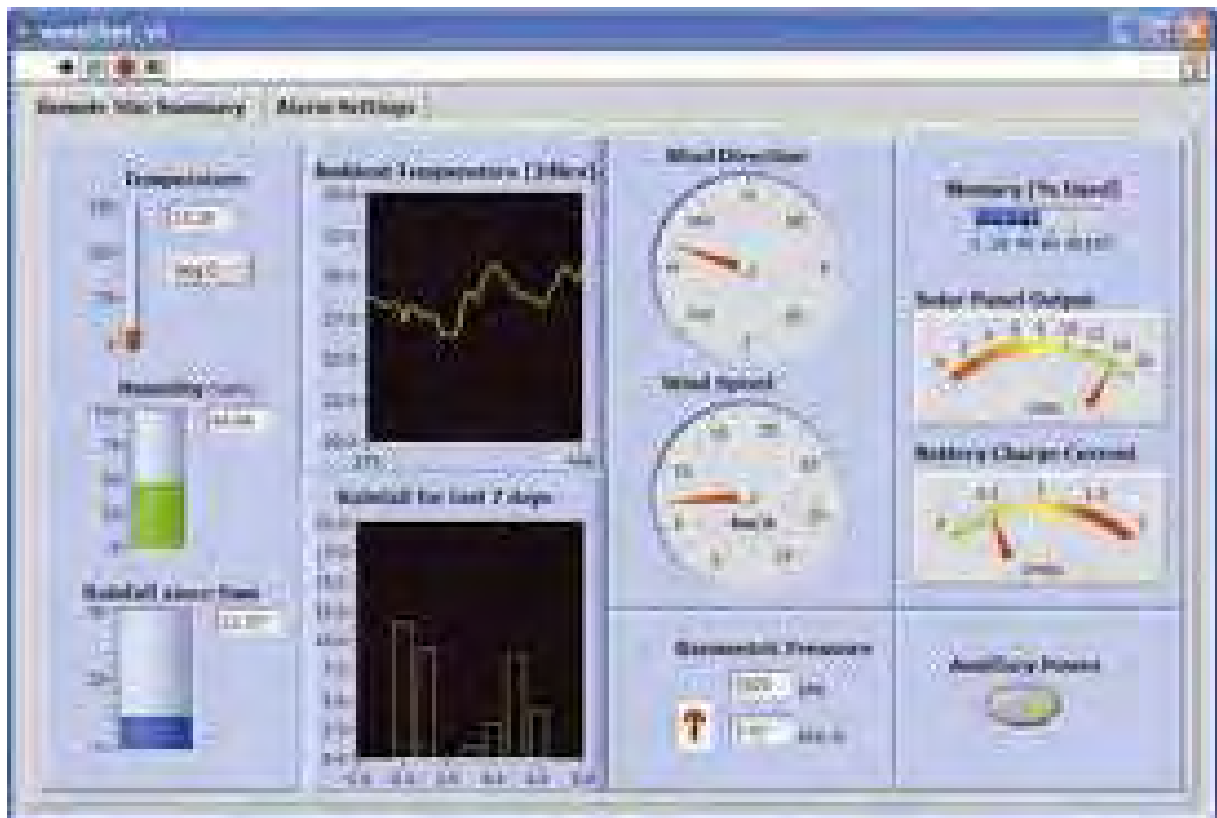


Figura2.10: Programa gráfico ejemplo en LabVIEW

La similitud existente entre los diagramas de flujo y los programas gráficos acorta la curva de aprendizaje asociada con lenguajes tradicionales basados en texto.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

El usuario puede determinar el comportamiento de los instrumentos virtuales conectando íconos entre sí para crear diagramas de bloques, que son notaciones de diseño naturales para ingenieros y científicos. Con un lenguaje gráfico se puede desarrollar sistemas más rápidamente que con lenguajes de programación convencionales mientras que conserva la potencia y flexibilidad necesarias para crear una variedad de aplicaciones.

### Conectividad y Control de Instrumentos en LabVIEW

LabVIEW posee bibliotecas listas para ser utilizadas con el objeto de integrar instrumentos autónomos, equipos de adquisición de datos, productos para el control de movimientos y de visión, instrumentos GPIB/IEEE 488 y serie RS-232 y PLCs, entre otros, lo cual permite construir una solución completa de medición y automatización. LabVIEW también tiene incorporadas las más importantes normas de instrumentación, tal como VISA, una norma que permite la operación entre instrumentos GPIB, serie y VXI; PXI y software y hardware basados en la norma PXI Systems Alliance CompactPCI; manejadores de instrumentos virtuales intercambiables IVI y VXIplug&play, que es un manejador para la norma que rige la instrumentación VXI.

### Medio Ambiente Abierto

Aunque LabVIEW provee las herramientas requeridas para la mayoría de las aplicaciones, también constituye un medio ambiente de desarrollo abierto. El cumplimiento de normas por parte del software se basa principalmente en la habilidad del paquete que el usuario elige para trabajar bien con otros sistemas de software y hardware de medición y control, y normas abiertas, que definen la capacidad de interactuar entre múltiples fabricantes. Seleccionando el software que cumple con estos criterios se asegura que la empresa y las aplicaciones aprovechen los productos ofrecidos por numerosos proveedores. Además, ajustándose a normas comerciales abiertas, usted reduce el costo total del sistema.

Un gran número de fabricantes de hardware y software desarrollan y mantienen centenares de bibliotecas de LabVIEW y manejadores de instrumentos que le ayudan a utilizar fácilmente sus productos con LabVIEW. Sin embargo, esa no es la única manera de proveer conectividad a las aplicaciones basadas en LabVIEW. Este software ofrece maneras simples de incorporar programas en ActiveX, bibliotecas dinámicas (DLLs) y bibliotecas compartidas de otras herramientas.

### Reducción de Costos y Preservación de la Inversión con LabVIEW



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

La instrumentación virtual con LabVIEW demuestra ser económica, no sólo por los reducidos costos de desarrollo sino también porque preserva la inversión del capital a lo largo de un extenso período. A medida que cambian sus necesidades, se puede fácilmente modificar los sistemas sin necesidad de adquirir nuevo equipamiento y crear bibliotecas enteras de instrumentación a costo menor que el correspondiente a un solo instrumento comercial tradicional.

### Plataformas Múltiples

La mayoría de los sistemas por computador utilizan alguna variante del sistema operativo Microsoft Windows; no obstante ello, existen otras opciones que ofrecen claras ventajas para ciertos tipos de aplicaciones. El desarrollo de sistemas operativos de tiempo real y embebido continúa creciendo rápidamente en la mayoría de las industrias a medida que la capacidad de cálculo es incorporada en paquetes más especializados y pequeños. Es importante minimizar las pérdidas resultantes del cambio hacia nuevas plataformas y la elección del software correcto para dicho objetivo es un factor clave.

LabVIEW minimiza esta preocupación ya que corre en Windows 2000, NT, XP, Me, 98, 95 y NT embebido así como también sobre Mac OS, Sun Solares y Linux. LabVIEW es independiente de la plataforma seleccionada: los instrumentos virtuales que usted pueda crear en una plataforma pueden ser transportados de manera transparente a cualquier otra plataforma LabVIEW simplemente abriendo el instrumento virtual.

Puesto que las aplicaciones de LabVIEW pueden transportarse entre plataformas, se puede asegurar que el trabajo de hoy será utilizable en el futuro. A medida que emerjan nuevas tecnologías computacionales, se pueden migrar fácilmente sus aplicaciones a las nuevas plataformas y sistemas operativos. Además, debido a que se pueden crear instrumentos virtuales que son independientes de la plataforma y transportarlo entre distintas plataformas, permite ahorrarse tiempo de desarrollo y otros inconvenientes relacionados con la portabilidad entre plataformas.

### Desarrollo Distribuido

Con LabVIEW se pueden desarrollar fácilmente aplicaciones distribuidas, aún entre diferentes plataformas. Con herramientas de servidores fáciles de usar, se pueden descargar rutinas que hacen un uso intensivo del procesador a otras máquinas para lograr una ejecución más rápida, o crear aplicaciones de monitoreo y control remoto. Una tecnología de servidores poderosa puede simplificar la tarea de desarrollar aplicaciones grandes y que requieran de múltiples



computadoras. Además de ello, LabVIEW incluye tecnologías normalizadas de redes, tales como TCP/IP e incorpora protocolos robustos de publicación y suscripción.

### Capacidades de Análisis

El software de la instrumentación virtual requiere de complejas herramientas de análisis y procesamiento de señales ya que la aplicación no se detiene justo cuando el dato es recogido. Usualmente, las aplicaciones de mediciones de alta velocidad en los sistemas de monitoreo y control de maquinaria requieren análisis de orden para lograr obtener datos de vibración precisos.

Además de las bibliotecas de análisis avanzado ya incluidas en LabVIEW, National Instruments provee software adicional, tal como el LabVIEW Signal Processing Toolset (Paquete de Herramientas para el Procesamiento de Señales LabVIEW) a fin de complementar las ofertas de análisis.

### Capacidades de Visualización

LabVIEW incluye un amplio conjunto de herramientas de visualización para presentar datos en la interface del usuario de la instrumentación virtual, tanto para gráficos continuos como también para visualización de gráficos 2D y 3D. El usuario puede reconfigurar de manera instantánea los atributos de la presentación de los datos, tales como: colores, tamaño de fuentes, tipos de gráfico y más, así como también efectuar rotación, enfoque (zoom) y desplazamiento dinámico en estos gráficos con el ratón. En lugar de programar gráficos y todos los atributos corrientes desde cero, se puede simplemente arrastrar y tirar estos objetos dentro de los paneles frontales de los instrumentos.

### Diseño e Investigación

En la investigación y el diseño, los ingenieros y científicos demandan capacidades de rápido desarrollo y realización de prototipos. Con los instrumentos virtuales se puede desarrollar rápidamente un programa, tomar mediciones desde un instrumento para ensayar un prototipo y analizar resultados, todo ello en una fracción del tiempo requerido para ejecutar ensayos con instrumentos tradicionales. Cuando usted necesita flexibilidad es esencial tener una plataforma ajustable y abierta, desde la computadora de mesa a los sistemas embebidos y redes distribuidas.



Los exigentes requerimientos de las aplicaciones de investigación y desarrollo (I&D) requiere una integración ininterrumpida de software y hardware. LabVIEW facilita la integración, tanto si se necesita comunicarse con instrumentos autónomos utilizando GPIB o adquirir de manera directa señales a la computadora con una placa de adquisición de datos y hardware de acondicionamiento de señales. Con los instrumentos virtuales también se puede automatizar un procedimiento de ensayo, eliminando la posibilidad de error humano y asegurando la consistencia de resultados al evitar introducir variables desconocidas o inesperadas.

### Ensayos de Desarrollo y Validación

Dada la flexibilidad y poder de los instrumentos virtuales, se pueden elaborar fácilmente complejos procedimientos de ensayo. En el campo de los ensayos automáticos de diseño de verificación, se pueden crear rutinas en LabVIEW e integrarlas con programas tales como el National Instruments TestStand, que ofrece capacidades de manejo de ensayos muy poderosas. El usuario puede desarrollar código dentro del proceso de diseño y luego insertar esos mismos programas dentro de herramientas funcionales para la validación, ensayo o la manufactura.

Los conceptos de software y hardware integrados de instrumentación virtual, herramientas modulares flexibles y la utilización de tecnologías comerciales se combinan para crear un marco de trabajo sobre el cual se puede completar rápidamente el desarrollo de los sistemas y también mantenerlos en el largo plazo. Puesto que la instrumentación virtual ofrece tantas opciones y capacidades en el desarrollo embebido tiene sentido que los desarrolladores de sistemas embebidos comprendan y reuean esas herramientas.

### Programación en LabVIEW. Sistemas de adquisición de datos.

El esquema general de una cadena de medida cuya misión es la adquisición de datos puede ser el siguiente:

*Figura2.11: Cadena de medida. Adquisición de datos*



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

Este esquema general o alguna de sus variaciones lo encontraremos en múltiples entornos de los que cabe destacar dos: uno será el de los procesos industriales y el otro será el de ensayos y test.

El entorno de este proyecto es el de ensayos y test.

A diferencia de un instrumento real, que podemos tener en cualquier laboratorio o planta de procesos, y que queda perfectamente definido por unos mandos de control y unos elementos de representación, un instrumento virtual estará ligado al concepto de software. Este software se ejecutará en un ordenador que tendrá alojado unos elementos hardware concretos, tarjetas de adquisición de datos (analógicos y digitales), tarjetas de interfaz con los buses de instrumentación y unos canales de control también analógicos y digitales.

El instrumento virtual permitirá manejar ese hardware mediante una interfaz gráfica de usuario (IGU) que se asemejará al panel de mandos de un osciloscopio real.

Mediante la representación en pantalla de los elementos gráficos de visualización y control que servirán de interfaz con el usuario, este observará los estados de las entradas seleccionadas en la pantalla e interactuará con las salidas directamente o mediante la ejecución de las rutinas programadas.

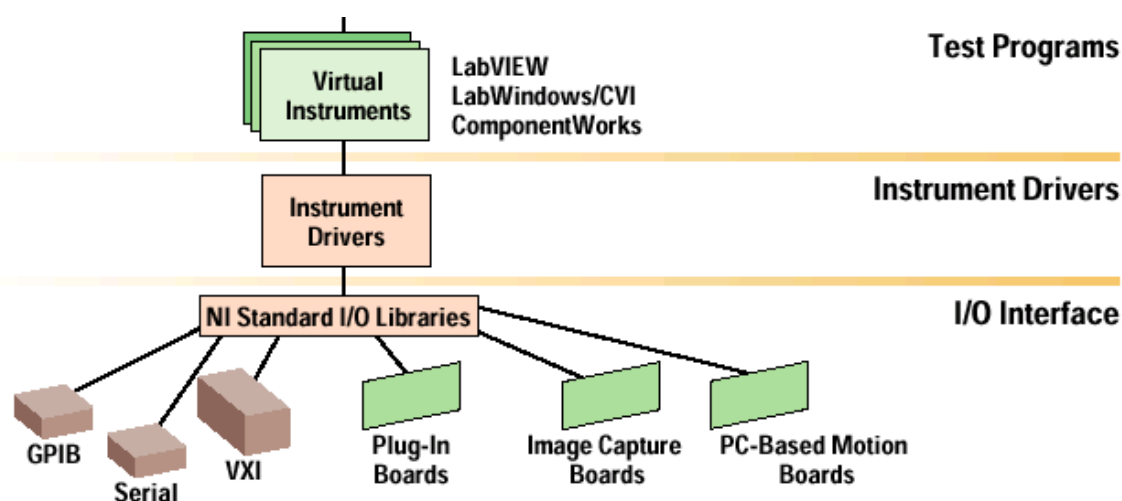


Figura2.12: Rutinas programadas en la ejecución de instrumentación virtual



Básicamente, el software se encargará de comunicar la interfaz de usuario del PC con el hardware de adquisición de datos dotando a la aplicación de la funcionalidad requerida, en este caso la funcionalidad del osciloscopio y analizador de espectro.

- Podemos realizar una separación de las capas o partes del software:
  - Programa de aplicación
  - -Controladores de dispositivo (drivers)
  - -Librerías de aplicación (API's)

### **Conclusión**

La instrumentación virtual está motorizada por la siempre creciente tecnología computacional que le ofrece al usuario el poder de crear y definir su propio sistema basado en un marco de trabajo abierto. Este concepto no sólo le asegura que su trabajo será utilizable en el futuro sino que también le provee la flexibilidad de adaptarlo y extenderlo a medida que cambian sus necesidades. LabVIEW provee herramientas poderosas y un medio ambiente de desarrollo familiar creado específicamente para el diseño de instrumentos virtuales. Éste es el porqué se ha elegido la utilización de LabVIEW como herramienta de desarrollo de éste proyecto.





## **Capítulo 3: Planificación de recursos y costes**

### **Introducción:**

El objetivo fundamental de la planificación del costo de producción, consiste en la determinación previa de los gastos indispensables para obtener un volumen dado de producción y entrega de cada tipo y de toda la producción de la empresa, con la calidad establecida. Asimismo, la magnitud del costo planificado se utiliza en:

La confección de los planes de las empresas, expresando la eficiencia de su actividad económico-productiva. La valoración de la efectividad económica de las diferentes medidas técnico-organizativas y de la producción en su conjunto. La valoración de la eficiencia económica obtenida en cada una de las áreas de la empresa (talleres, establecimientos, brigadas). Los análisis de eficiencia a tomar en cuenta al formar los precios de cualquier tipo de producción.

El costo planificado se determina en la empresa mediante los cálculos técnico-económicos de la magnitud de los gastos para la fabricación de toda la producción y de cada tipo de artículo que compone el surtido de la misma.

En la medición de la efectividad del plan y en su ejecución, se emplean los indicadores del costo por unidad de producción en las producciones comparables y costo por peso de producción bruta y mercantil, para el nivel de actividad total de la empresa.

### **Valoración de tareas e hitos del proyecto**

Nombre de tarea	Duración	Comienzo	Fin
-----------------	----------	----------	-----



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

<b>Requerimientos del sistema</b>	<b>4,88 días</b>	<b>lun 02/01/12</b>	<b>vie 06/01/12</b>
Investigación Inicial	1 día	lun 02/01/12	lun 02/01/12
Identificación de necesidades	1 día	mar 03/01/12	mar 03/01/12
Esquema con los datos recogidos y puestas en conjunto	1 día	mié 04/01/12	mié 04/01/12
Plasmo de los datos en documentación	1 día	jue 05/01/12	jue 05/01/12
<b>Requerimientos Hardware</b>	<b>10,13 días</b>	<b>lun 02/01/12</b>	<b>vie 13/01/12</b>
Estudio detallado del manual de la tarjeta de adquisición de datos DAQ-PAD 6020E	5 días	lun 02/01/12	vie 06/01/12
Plasmo de los datos en documentación	2 días	jue 12/01/12	vie 13/01/12
<b>Requerimientos Software</b>	<b>21,88 días</b>	<b>jue 12/01/12</b>	<b>vie 10/02/12</b>
Estudio detallado de LabVIEW para el desarrollo software en instrumentacion virtual	4,88 días	jue 12/01/12	mié 18/01/12
Estudio de: LabVIEW Intermediate I: Successful Development Practice Course Manual	6,88 días	mié 18/01/12	vie 27/01/12
Estudio de: LabVIEW Intermediate II: Perfoncance and Connectivity Course Manual	4,88 días	vie 27/01/12	vie 03/02/12
Estudio de: Introducción a LabVIEW, dto. SEC UPM	4,88 días	vie 03/02/12	vie 10/02/12
<b>Diseño preliminar, diagramas de bloques</b>	<b>7 días</b>	<b>lun 13/02/12</b>	<b>mar21/02/12</b>
Generador de funciones	2 días	lun 13/02/12	mar 14/02/12
Analizador de espectro	3 días	lun 13/02/12	mié 15/02/12
Osciloscopio	2 días	lun 13/02/12	mar 14/02/12
Puesta en común	2 días	jue 16/02/12	vie 17/02/12



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

<b>Codificación del Software</b>	<b>14,13 días</b>	<b>mié 22/02/12</b>	<b>lun 12/03/12</b>
Generador de funciones	10 días	mié 22/02/12	mar 06/03/12
Analizador de espectro	14 días	mié 22/02/12	lun 12/03/12
Osciloscopio	7 días	mié 22/02/12	jue 01/03/12
<b>Depuración del código software</b>	<b>16,13 días</b>	<b>vie 02/03/12</b>	<b>vie 23/03/12</b>
Generador de funciones	13,13 días	mié 07/03/12	vie 23/03/12
Analizador de espectro	9,13 días	lun 12/03/12	vie 23/03/12
Osciloscopio	6,13 días	vie 02/03/12	vie 09/03/12
<b>Integración y pruebas</b>	<b>10,13 días</b>	<b>lun 26/03/12</b>	<b>vie 06/04/12</b>
Integración sistema completo	4,13 días	lun 26/03/12	jue 29/03/12
DAQPad 6020-E	4,13 días	lun 26/03/12	vie 30/03/12
Pruebas de sistema completo	6 días	vie 30/03/12	vie 06/04/12

Figura3.1: Tabla de tareas y esquema temporal

### Costes para la empresa

#### Elección de licencia de LabVIEW

National Instruments ofrece los siguientes paquetes software para empresas. Cada paquete ofrece una serie de librerías, las cuales están disponibles para explotación de las empresas que



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

paguen la licencia, por un total de un año, dos años, cinco años o diez años. Los precios de licencia por un año de todos los paquetes software que ofrece National Instruments se presentan a continuación:

Development Systems   Add-Ons   [View Cart](#) | [View Quote](#)





LabVIEW Base	LabVIEW Full	LabVIEW Professional	NI Developer Suite
			
starting at €1299	starting at €2799	starting at €4649	starting at €5049
<a href="#">Add to Cart</a> <a href="#">Instant Quote</a>	<a href="#">Add to Cart</a> <a href="#">Instant Quote</a>	<a href="#">Add to Cart</a> <a href="#">Instant Quote</a>	<a href="#">Customize</a>
<b>Basic I/O</b> Hardware I/O integration and graphical UI development <a href="#">Learn More</a>	<b>Analysis</b> More than 850 analysis and math functions as well as event-driven programming <a href="#">Learn More</a>	<b>Deploy</b> Application distribution (.exe) and source code control <a href="#">Learn More</a>	<b>Best Value</b> LabVIEW Professional bundled with popular add-ons (up to 74% savings) <a href="#">Learn More</a>

Figura3.2: Licencias disponibles de LaBVIEW

Cada paquete ofrece una serie de características:



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

## Universidad Politécnica de Madrid

Features - Expand All	LabVIEW Base starting at €1,299	LabVIEW Full starting at €2,799	LabVIEW Professional starting at €4,649	NI Developer Suite starting at €5,049
Operating System	Windows only	Windows, Linux, Mac	Windows, Linux, Mac	Windows only
Customize User Interfaces	●●●●	●●●●	●●●●	●●●●
Hardware I/O Integration	●●●●	●●●●	●●●●	●●●●
▶ Standard Service Program (SSP)	●●●●	●●●●	●●●●	●●●●
▶ Compatibility with Add-On Software	○●●●	●●●●	●●●●	●●●●
▶ Mathematics	○○●●	●●●●	●●●●	●●●●
▶ Executables and Distribution	○○●●	○●●●	●●●●	●●●●
▶ Programming Environment	○○●●	○●●●	●●●●	●●●●
▶ Read, Write, and Share Data	○○●●	○●●●	○●●●	●●●●
▶ Software and Code Integration	○○●●	○●●●	○●●●	●●●●
▶ Signal processing and control	○○○○	○●●●	○●●●	●●●●

<sup>1</sup>Add-ons provide all functionality for a feature, but you must purchase them separately.

Legend ○○○○ No functionality ○○●● Little functionality ○●●● Most functionality ●●●● All functionality

Figura3.3: Características de cada una de las licencias disponibles de LabVIEW



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

### Add to Cart

✓ **LabVIEW Professional** has successfully been added to your cart

#### Need Add-Ons?

Please customize your selection below. Your price will update for changes to service.

Integrating with Deployment Hardware  
Signal Processing, Analysis, and Connectivity  
Control and Simulation  
Data Management, Logging, and Reporting  
Development Tools and Validation  
Application Distribution

Add-Ons

View Cart

Close

*Figura3.4: Compra de la licencia de LabVIEW*

En total, explotar la licencia por un año costaría a la empresa 4649 €.





# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

## Nóminas

FRAILE GONZALEZ, JESUS FIDEL  
CL YUNCOS 12  
45200 ILLESCAS  
TOLEDO

160 300004

EMPRESA	FLIR NETWORKED SYSTEMS, S.L.U.	TRABAJADOR/A	FRAILE GONZALEZ, JESUS FIDEL
Domicilio:	AV. BRUSELAS, 15	N.I.F.:	47288386H
C.I.F.:	B83333492	Número libro de Matrícula:	
Código de la cuenta de cotización a la Seguridad Social:	28/1844947-01	Núm. afiliación a la Seguridad Social:	28/13641347-80
		Categoría o grupo profesional:	Ingeniero
		Cod.CT:	100G, Cotiz:
		F antigüedad:	17/10/2011

Periodo de liquidación: MENS del 1 de ABRIL al 30 de ABRIL de 2012 Total Días: ( 30 )

I. DEVENGOS		TOTALES
1. Percepciones salariales		
Salario base	1540 Euros	
Horas extraordinarias		
Gratificaciones extraordinarias		
Salario en especie		
Complementos salariales		
	1540 Euros	
2. Percepciones no salariales		
Indemnizaciones o sueldos		
Prestaciones e indemnizaciones de la Seguridad Social		
del al		80 Euros
del al		
Indemnizaciones por traslados, suspensiones o despidos		
Otras percepciones no salariales		
A. TOTAL DEVENGADO		234 Euros

II. DEDUCCIONES	
1. Aportación del trabajador a las cotizaciones a la Seguridad Social y conceptos de recaudación conjunta:	
Contingencias comunes	%
Desempleo	%
Formación Profesional	%
Horas extraordinarias	%
Fuerza mayor	%
Riesgo horas extraordinarias	%
TOTAL APORTACIONES	
2. Impuesto sobre la renta de las personas físicas	
3. Anticipos	
4. Valor de los productos recibidos en especie	
5. Otras deducciones	
B. TOTAL A DEDUCIR	
LIQUIDO TOTAL A PERCIBIR (A-B)	
234 Euros	
1306 Euros	

30 de ABRIL de 2012  
RECIBI

CUENTA

DETERMINACIÓN DE LAS BASES DE COTIZACIÓN A LA SEGURIDAD SOCIAL Y CONCEPTOS DE RECAUDACIÓN CONJUNTA Y DE LA BASE SUJETA A RETENCIÓN DEL I.R.P.F.

1. Base de cotización por contingencias comunes	2. Base de cotización por contingencias profesionales (A.T. y E.P.) y conceptos de recaudación conjunta (Desempleo, Formación Profesional, Fondo de Garantía Salarial)
Remuneración mensual	
Pro rata pagas extraordinarias	
Base Incapacidad temporal	
TOTAL	154 Euros

Figura3.5: Nómina de cada ingeniero





## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

El sueldo de cada ingeniero en bruto, para la empresa será de 1540 Euros, más el 38 % del bruto para la seguridad social. En total, cada mes la empresa deberá invertir 2125 Euros por ingeniero y por mes.

	Enero	Febrero	Marzo	Total
Ingeniero 1	2125€	2125€	2125€	6375€
Ingeniero 2	2125€	2125€	2125€	6375€
Ingeniero3	2125€	2125€	2125€	6375€
				19125€

Figura3.6: Tabla de sueldos en bruto de los ingenieros del proyecto

Coste del proyecto:

	Enero	Febrero	Marzo	Total
Ingeniero 1	2125€	2125€	2125€	6375€
Ingeniero 2	2125€	2125€	2125€	6375€
Ingeniero 3	2125€	2125€	2125€	6375€
Licencia LabVIEW Professional	-	-	-	4649€
Gastos Varios:				
Luz,agua,material de oficina, limpieza, teléfono, local,	1200€	1200€	1200€	3600€



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

mantenimiento informático.....

**Total coste proyecto de Instrumentación Virtual:**

**27374 €**

*Figura3.7: Tabla de gastos totales, en bruto*



**Escuela Universitaria de Ingeniería  
Técnica de Telecomunicación**

**Universidad Politécnica de Madrid**



## **Capítulo 4: Fundamentos teóricos**

### **Introducción:**

El propósito de este capítulo es proporcionar al lector, un recordatorio o refresco de conceptos sobre teoría de señal, adquisición de datos, muestreo de señales analógicas, analizador de espectro, transformada de Fourier, interpolación y filtros para mejorar la pureza espectral y algunas herramientas para la reducción de ruido.

En caso de que el lector no sea entendido en la materia, este capítulo también puede servir como un tutorial de aprendizaje para este tipo de sistemas. Se intentará explicar con ejemplos todos los fundamentos, para darle la posibilidad al lector que, habiendo estudiado con anterioridad estos conceptos, o siendo un aficionado en la materia, comprender los siguientes capítulos.

En los siguientes capítulos, en concreto, los capítulos de Ingeniería del Hardware e Ingeniería del Software, es donde se llevará a cabo el diseño de la aplicación. Con ayuda de estos fundamentos teóricos, se debería llegar a un entendimiento completo de lo que se llevará a cabo posteriormente.

### **Introducción a la adquisición de datos:**

La adquisición de datos trata de unificar las señales procedentes de las fuentes de medida, digitalización, almacenamiento, análisis y muestra de resultados en un PC, en un sistema o varios sistemas capaces de realizar todo lo anteriormente mencionado. Actualmente, la



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

tecnología ofrece multitud de posibilidades y flexibilidad a la hora de elegir tipos de sistemas de adquisición de datos. Por poner unos ejemplos, en el mercado hay sistemas PCI, PXI, PCI Express, PXI Express, PCMCIA, USB, Wireless o sistemas de adquisición para test, medida y aplicaciones de automoción via Ethernet.

Para construir un sistema de adquisición básico, serán necesarios los siguientes cinco componentes:

- Sensor / Transductor
- Señales
- Acondicionamiento de señal
- Hardware DAQ
- Aplicación Software (Drivers)

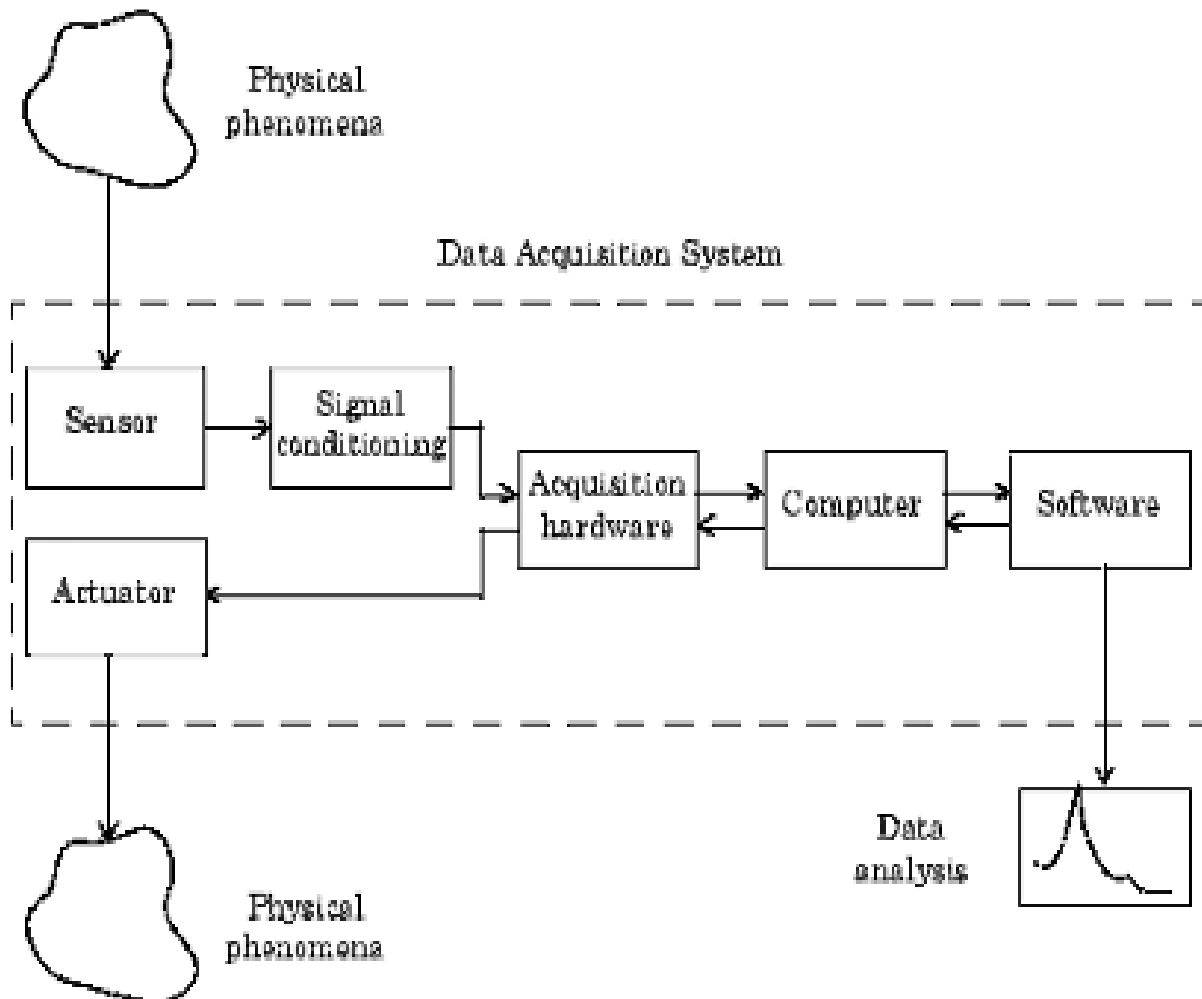


Figura 4.1 Sistema de adquisición de datos básico

## Sensor / Transductor

Toda adquisición de datos comienza con un fenómeno físico (temperatura, humedad, presión, etcétera), el cual queremos medir. Un sensor o transductor es un dispositivo que genera una señal eléctrica, como una tensión o una corriente, en función del fenómeno físico que se está midiendo.

## Señales



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

Los sensores apropiados para cada fenómeno físico general señales medibles, pero se ha de tener en cuenta que diferentes señales medibles necesitan ser medidas de diferente manera, por tanto, es importante entender los diferentes tipos de señales y sus correspondientes características, las cuales se expondrán a continuación.

Las señales se pueden catalogar en dos tipos, señales analógicas y señales digitales.

- Señales analógicas: la señal analógica es la que puede tomar cualquier valor en función del tiempo. Señales analógicas pueden ser, por ejemplo, tensión, corriente, presión, temperatura, sonido, etcétera. Las tres principales características de las señales analógicas son el nivel de señal, la forma y la frecuencia.

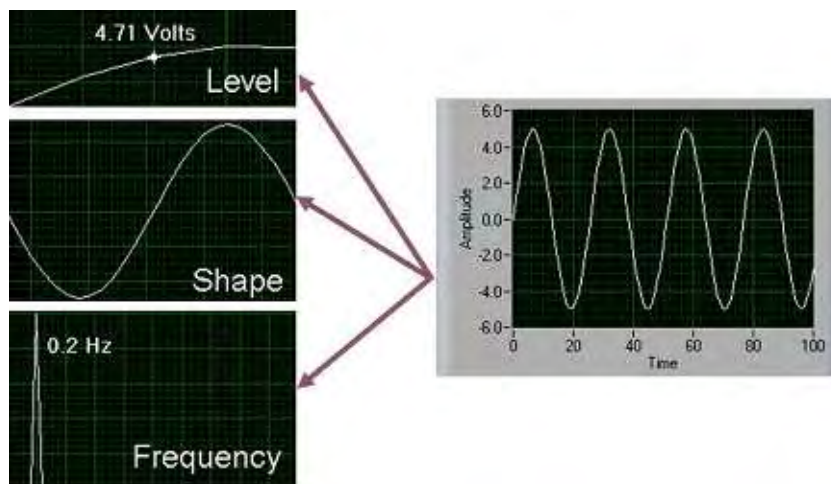


Figura 4.2 Señales analógicas

- Nivel de señal: Como las señales analógicas pueden tomar cualquier valor en el tiempo, el nivel de señal nos da la información de ese valor medido de señal analógica.
- Forma: algunas señales son nombradas conforme a como la señal cambia respecto al tiempo. Las formas más comunes pueden ser la señal seno, señal cuadrada y triangular. La forma de la señal analógica es importante, ya que nos permite analizar más en profundidad la señal midiendo sus valores de pico, valores de continua, pendientes de subida y/o bajada, etcétera.





- Frecuencia: todas las señales analógicas pueden ser catalogadas por su frecuencia. A diferencia del nivel de señal o la forma, no se puede analizar directamente la frecuencia de la señal con un sistema de adquisición. Para poder analizar la frecuencia de la señal, debemos ayudarnos de algoritmos software. Este análisis software de la frecuencia, comúnmente se lleva a cabo mediante la transformada de Fourier. Se ha de tener en cuenta que, cuando la frecuencia es una pieza clave en el contenido de la información de la señal, el sistema de adquisición debe ser preciso y debe de tener la suficiente velocidad de muestreo para no perder información en la adquisición, tema que se tratará posteriormente, en este capítulo (Teorema de Nyquist).
- Señales digitales: las señales digitales solo pueden tomar dos posibles valores, o niveles, en función del tiempo: alto o bajo. Las dos principales características de las señales digitales que se pueden medir son el estado y la velocidad.

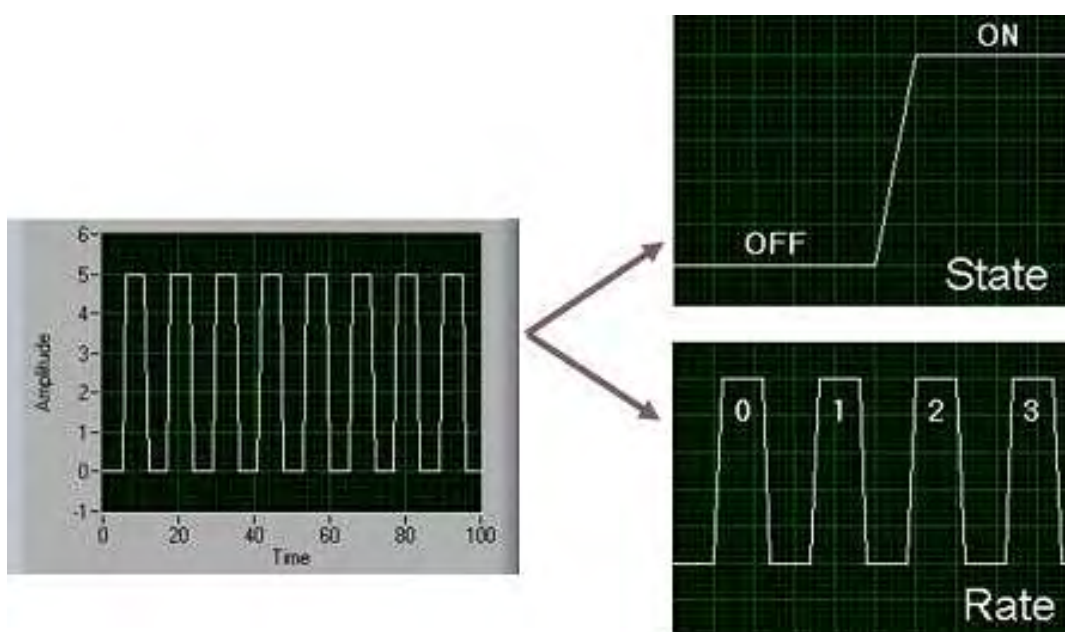


Figura 4.3 Señales digitales



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

- Estado: el estado es simplemente el nivel de señal, referido comúnmente como alto o bajo, encendido o apagado, abierto o cerrado.
- Velocidad: la velocidad de la señal digital define como la señal digital cambia de estado respecto al tiempo. A diferencia de la frecuencia, la velocidad de una señal digital nos informa de cada cuanto tiempo ocurre una porción de señal. No es necesario un algoritmo software para determinar la velocidad de una señal digital.

### Acondicionamiento de señal

Algunas veces, los transductores generan señales difíciles o peligrosas de medir directamente con un dispositivo de adquisición de datos. Por ejemplo cuando tratamos con altas tensiones o corrientes, o altas temperaturas. En estos casos, es esencial un acondicionamiento de la señal. El acondicionamiento de señal transforma la señal difícil de medir, o peligrosa, en una señal preparada para ser medida con precisión y garantizando la seguridad del sensor para realizar de forma segura y correcta la medida. El acondicionamiento de señal generalmente lo realiza un hardware para uso de aplicaciones como amplificación, atenuación, aislamiento, multiplexación...



### Modular Signal Conditioning Systems



### Integrated Signal Conditioning Devices



Figura 4.3 Sistemas de acondicionamiento de señal

Interfaz Hardware de adquisición de datos: Hardware DAQ



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

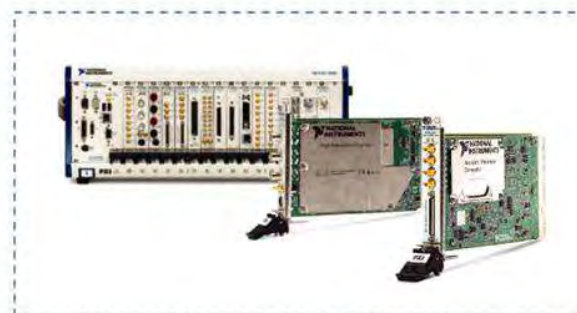
La interfaz hardware de adquisición de datos hace de puente entre el PC y el mundo real. Su función primaria como dispositivo es digitalizar la información analógica que le entra, para que un sistema digital pueda entender la información, para después procesarla, analizarla, etcétera. Los DAQ más comúnmente usados son los DAQ de escritorio, pero hay de muchos tipos para diferentes aplicaciones y situaciones. En el siguiente capítulo se hará una explicación exhaustiva del DAQ que se utilizará en este proyecto. Solo adelantar que se trata de un DAQ de escritorio, de National Instruments.



Distributed



Desktop



PXI: Rugged and Modular



Portable

Figura 4.4 Sistemas utilizados en adquisición de datos



## Software

El Software consigue recoger la información de la interfaz hardware del PC para llevar a cabo una adquisición completa de datos, análisis y representación de los mismos. El análisis en el caso concreto de este proyecto, será mayoritariamente un análisis espectral. Hay dos capas software en este tipo de sistemas, el driver y la capa de aplicación.

## Driver Software

El driver, en esencia, es la capa de software para que un usuario o programador consiga comunicarse fácilmente con el hardware, sin que la comunicación sea difícil ni incómoda a nivel de registros o comandos difíciles de memorizar para acceder a las funciones hardware.

## Aplicación Software

La capa de aplicación es la capa en la cual se desarrolla un medio en el cual se construyen funcionalidades concretas para un determinado fin. También puede usarse esta capa para configurar la interfaz hardware. La capa de aplicación software, además, añade funciones de análisis y representación para el driver software.

### **Fundamentos básicos de muestreo analógico:**



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

Para entender correctamente el muestreo analógico, el cual realizará la tarjeta DAQ como se explicará posteriormente en el capítulo de Ingeniería del Hardware, se ha de entender primero que es el ancho de banda.

Ancho de banda se define como la medida de un circuito o canal de transmisión para transmitir una señal sin atenuación significativa sobre un rango de frecuencias.

El ancho de banda es medido entre los puntos de frecuencia inferior y superior, donde la amplitud de la señal cae a -3dB por debajo de la frecuencia de paso de banda.

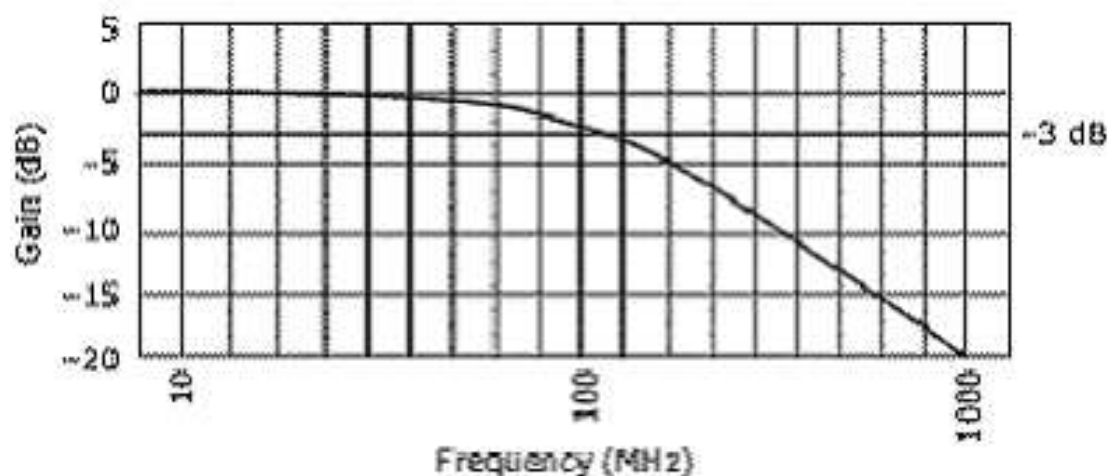


Figura 4.4 Diagrama de Bode de la respuesta típica de un digitalizador de 100MHz.

Por debajo de esos -3dB la señal empieza a atenuarse, como podemos ver en el siguiente ejemplo:



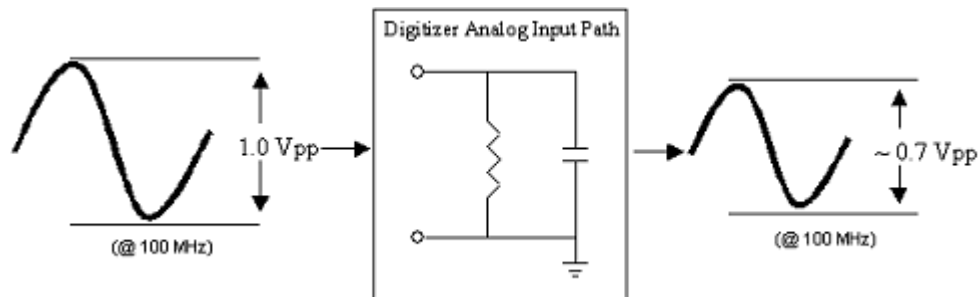


Figura 4.4 La señal de 100MHz se atenúa al pasar por el digitalizador, la tensión Vpp cae de 1Vpp a 0.7Vpp.

### Error teórico de medida de amplitud de una señal analógica

Se recomienda siempre que el ancho de banda de un sistema de medida sea de tres a cinco veces la máxima frecuencia de la señal que a medir, para cometer un error en la medida de la amplitud prácticamente despreciable. Este error de amplitud puede calcularse teóricamente con la siguiente fórmula:

$$\text{Error (\%)} = \left(1 - \frac{R}{\sqrt{1+R^2}}\right) * 100$$

Donde R es el ratio, en el caso anterior, del digitalizador, que a su vez es igual a

$R = \text{Ancho de banda} / \text{Frecuencia de entrada}.$

Otro concepto importante relacionado con el ancho de banda es el tiempo de subida de la señal de entrada al sistema. Este tiempo es el que tarda la señal en recorrer desde el 10% al 90% su valor de amplitud, y es inversamente proporcional al ancho de banda.



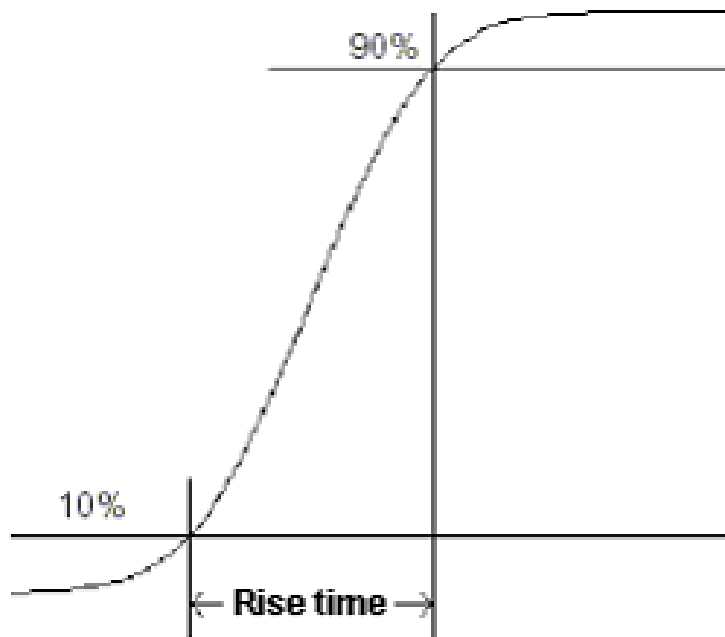


Figura 4.5 Tiempo de subida de la señal de entrada de un sistema.

## Velocidad de muestreo

La velocidad de muestreo de los datos no está directamente relacionada con el ancho de banda. Depende de la velocidad en la cual el convertidor analógico a digital convierte la señal de entrada analógica en una ristra de valores en código binario digital.

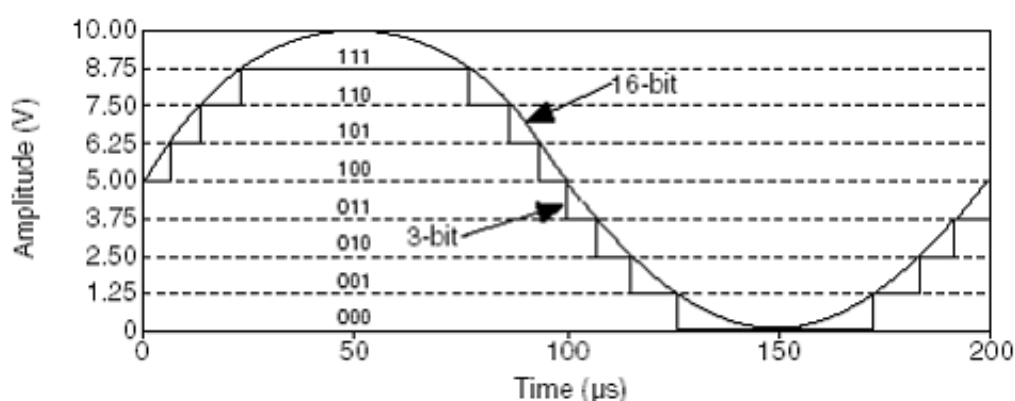


Figura 4.6 Código binario de una señal digital

### Teorema de Nyquist

La velocidad de muestreo, o frecuencia de muestreo, debe ser al menos dos veces la frecuencia mayor de la señal a medir. En caso de que la frecuencia de muestreo sea menor, se producirá llamado efecto aliasing, explicado a continuación.

La pregunta importante aquí es: ¿A qué velocidad se debe muestrear la señal que quiero medir?

La siguiente figura muestra el efecto de varias velocidades de muestreo. En el caso A, la señal sinusoidal de frecuencia ' $f$ ' es muestreada a la misma frecuencia ' $f$ '. En la forma de onda reconstruida aparece una componente continua. Sin embargo, si aumentamos la frecuencia de muestreo a la de Nyquist, la forma de onda reconstruida posee la misma frecuencia que la señal original, pero es triangular. Contra mayor sea la frecuencia de muestreo con respecto a la de Nyquist, más exacta será la reproducción de la señal original por la señal digitalizada.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

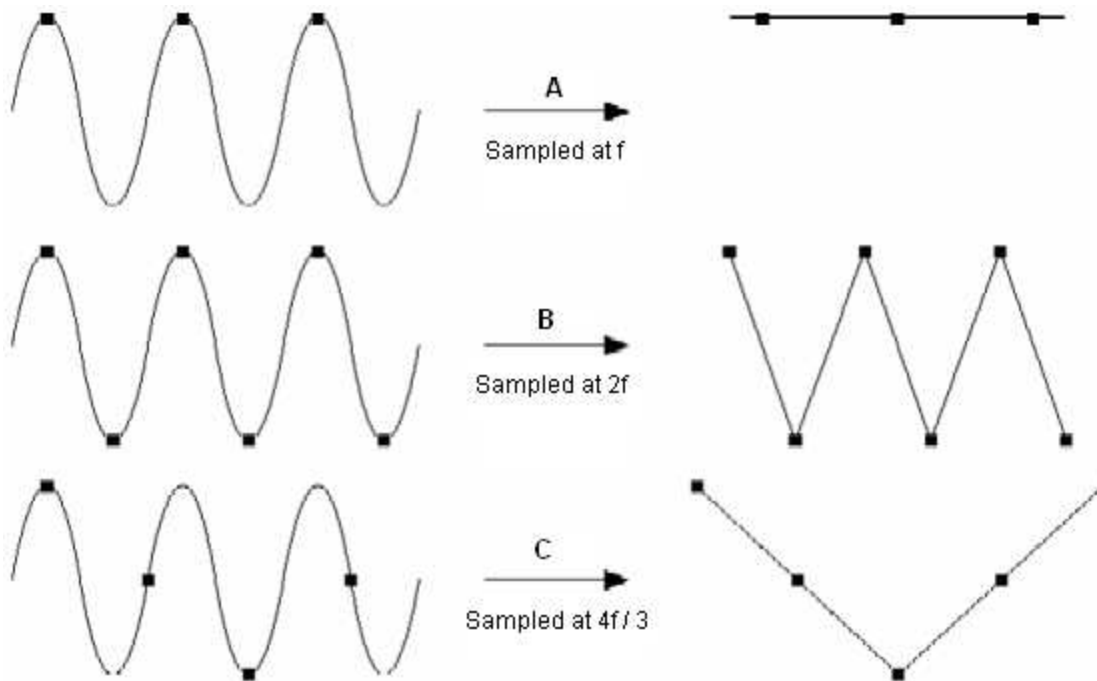


Figura 4.7 Efecto de varias velocidades de muestreo al muestrear una señal de frecuencia  $f$ .

Si la señal es muestreada a una velocidad menor que dos veces la frecuencia de Nyquist, una componente inesperada, la cual es una componente en frecuencia más pequeña que la de la señal original, aparece en la señal muestreada. Este fenómeno se denomina efecto Aliasing.

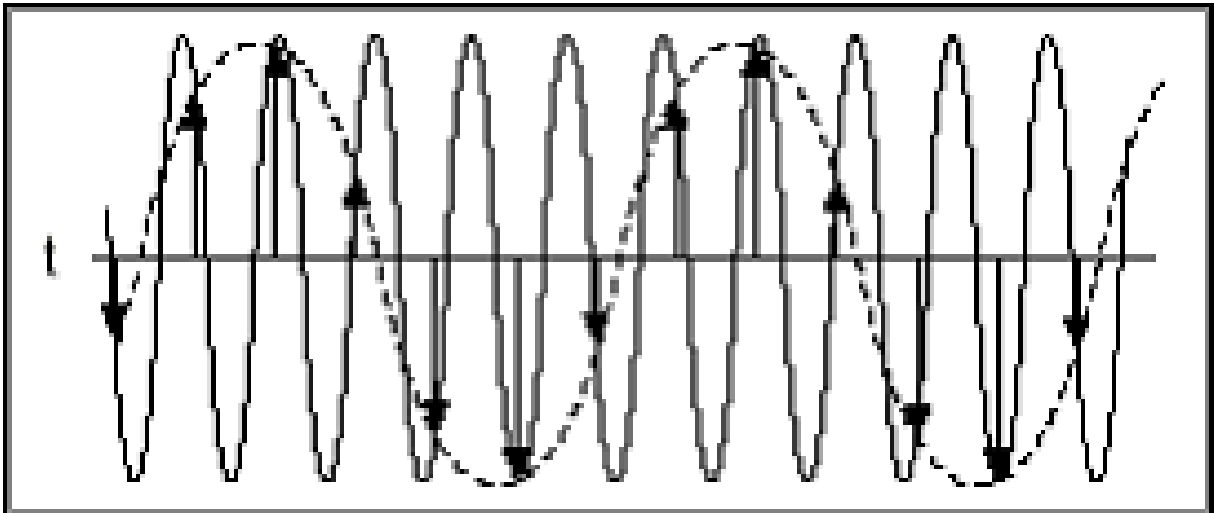


Figura 4.8 Efecto Aliasing

Las señales del mundo real a menudo contienen componentes de frecuencia que se encuentran por encima de la frecuencia de Nyquist.

Estas frecuencias son erróneamente añadidas a la componentes en frecuencia de la señal que se muestrea, produciendo distorsiones en los datos muestreados. En los sistemas en los que desea realizar mediciones precisas con datos muestreados, la tasa de muestreo debe ser suficientemente alta (alrededor de cinco a diez veces la componente de mayor frecuencia de la señal) para evitar el efecto aliasing. Otra opción sería la utilización de un filtro anti-aliasing (un filtro paso bajo que atenúa las frecuencias en la señal de entrada que son mayores que la frecuencia de Nyquist). La salida del filtro anti-aliasing debería estar conectada a la entrada del ADC para restringir el ancho de banda de la señal de entrada para satisfacer los criterios de muestreo.



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

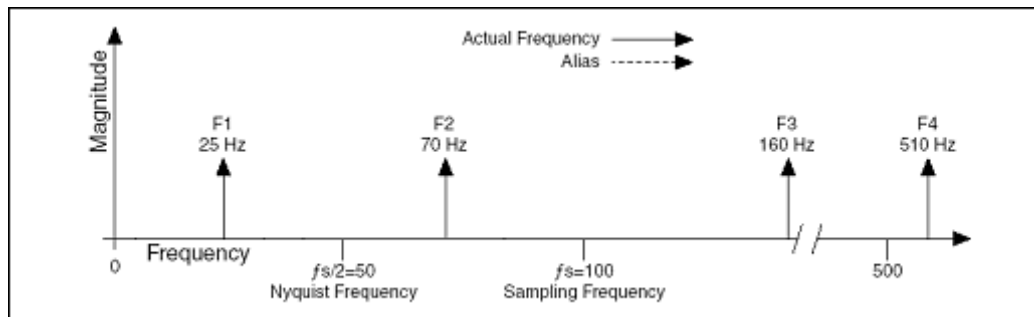


Figura 4.6 Ejemplo de componentes en frecuencia que aparecen en las señales del mundo real

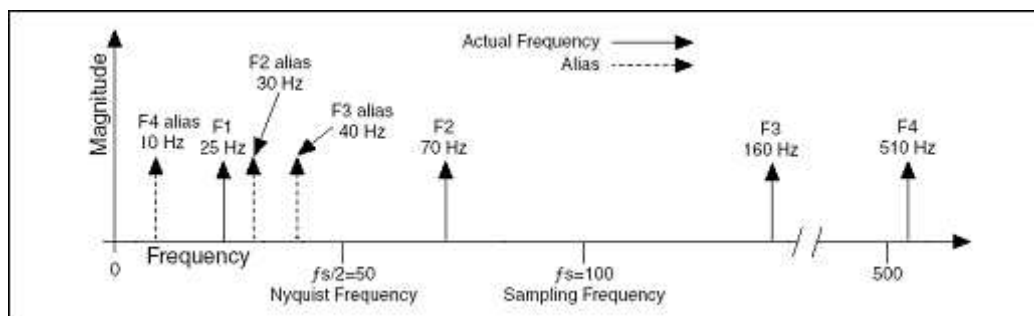


Figura 4.9 Componentes en frecuencia de la señal original junto a las componentes en frecuencia que aparecen debido al efecto aliasing en la señal muestreada.

## Cuantificación



Se define como el proceso de conversión de una señal analógica a una representación digital de ésta. Se lleva a cabo mediante un convertidor A/D. Si pudiésemos convertir las conversiones analógicas en un flujo de datos digitales conforme se van realizando las conversiones, aprovecharíamos esta cualidad de los PC con ayuda de un determinado software para poder realizar cualquier manipulación, procesamiento o estudio de las señales analógicas del mundo real, para después volver a reconstruir la señal manipulada o procesada otra vez en el dominio analógico si fuese necesario.

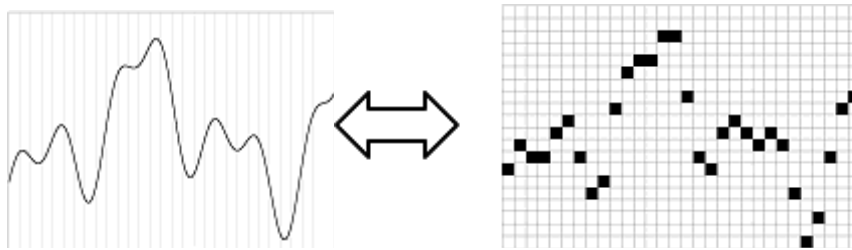


Figura 4.10 Cuando convertimos señales del dominio analógico al digital, los valores de señal son tomados como valores discretos en el tiempo.

La resolución temporal que tenemos está limitada por la máxima velocidad de muestreo del ADC. Incluso si pudiésemos ir aumentando la velocidad de muestreo de forma permanente, nunca podríamos llegar a realizar una reconstrucción perfecta de la señal. Lo mismo ocurre con los valores de amplitud, los cuales están limitados por la resolución de los valores discretos que el ADC puede sacar en su salida. Como se puede observar en la siguiente figura, contra mayor sea la resolución del ADC, mayor será la precisión con la que la señal será convertida, pero siempre habrá una cierta incertidumbre entre el valor digital y el valor analógico. Esta incertidumbre es llamada 'error de cuantificación', el cual depende en su mayor medida del número de bits del convertidor, pero también del ruido, las no linealidades y otros errores.



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

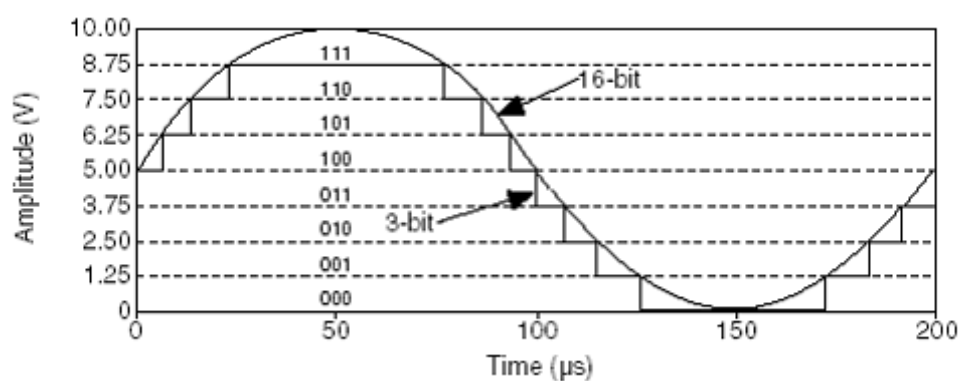


Figura 4.11 Imagen digital de una onda senoidal a cinco kiloHertzios obtenidos con un DAC de 3 bits

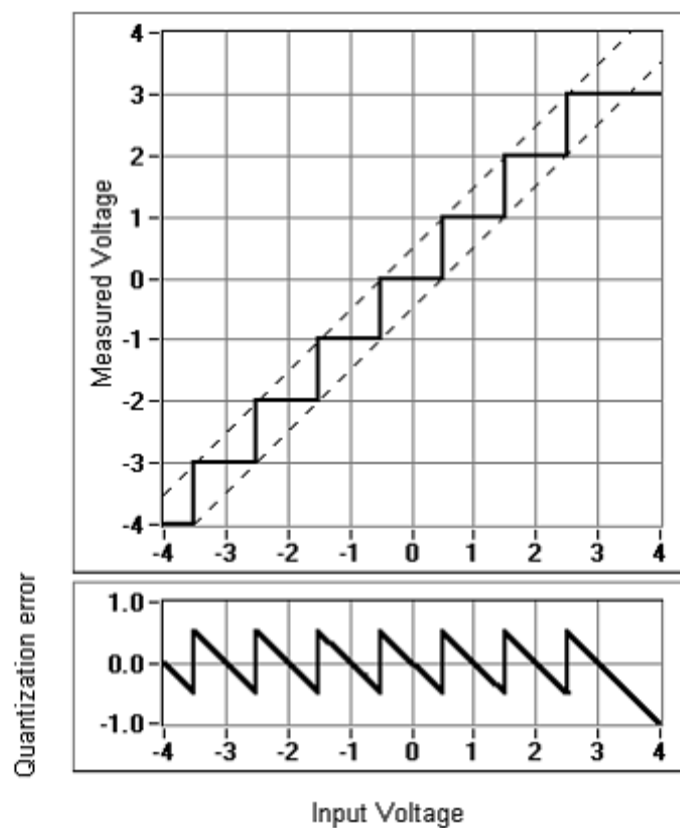


Figura 4.12 Error de cuantificación con el DAC de 3 bits





A continuación se muestran dos ejemplos de adquisición de señal con dos convertidores de distinta resolución. Estos ejemplos muestran, en ambas adquisiciones de una señal analógica de 2.5 voltios de rango usando un convertidor de 14 bits (16,384 escalones discretos de tensión), y un convertidor de 8bits (256 escalones discretos de tensión), la conversión digital de ambas señales analógicas.

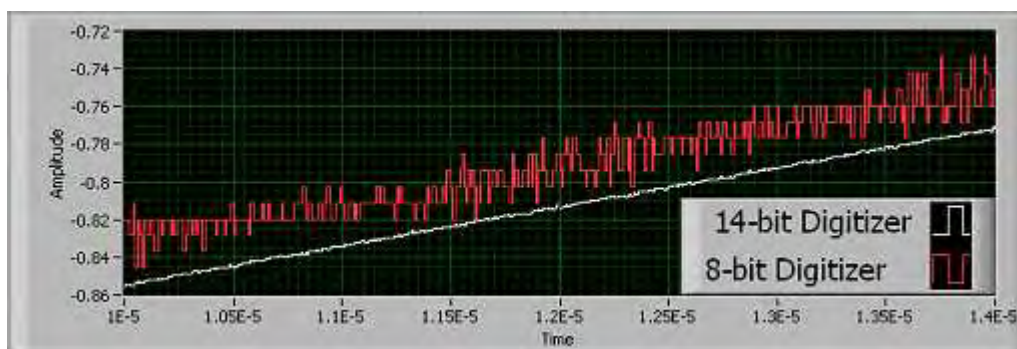


Figura 4.12 Convertidor de 8 bits contra convertidor de 14 bits.

**Especificaciones dinámicas hardware, para el generador de funciones y el analizador de espectro:**

Visión general



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

Para un generador de señal ideal, la generación de una onda sinusoidal analógica pura produciría un pico único en el dominio de la frecuencia. Sin embargo, los generadores de señal son dispositivos digitales que fundamentalmente sólo son capaces de aproximar la señal analógica ideal. Como resultado, es importante determinar la precisión de esta caracterización utilizando especificaciones dinámicas.

Si bien las especificaciones, tales como la resolución de bits y frecuencia de muestreo se utilizan para diferenciar los generadores de señal, estos factores por sí solos no pueden plenamente caracterizar el rendimiento de un generador de señal. Por lo tanto, este apartado se centra en las especificaciones dinámicas como , el rango dinámico libre de espurios SFDR, la distorsión armónica (THD), la relación señal-ruido y distorsión (SINAD), el número efectivo de bits (ENOB), la distorsión de intermodulación (IMD), la densidad de ruido y el ruido de fase para caracterizar mejor rendimiento de un generador de señal.

### Respuesta en frecuencia

Bajo condiciones ideales, la amplitud de un estímulo o señal en un generador de funciones se mantiene constante incluso a medida que aumenta la frecuencia. Sin embargo, debido a las características no ideales de los componentes de un generador de señal, la amplitud de cualquier señal dada puede variar ligeramente en función de su frecuencia. El ancho de banda de un generador de señal dado es la frecuencia a la cual la señal de salida es atenuada 3 dB con respecto a la amplitud de una continua o señal de baja frecuencia. De nuevo, el ancho de banda de una fuente de señal está limitado por el diseño de amplificador de salida o por los filtros en el circuito de salida analógica. Ancho de banda es uno de los factores que determinan el generador de frecuencia máxima de salida.

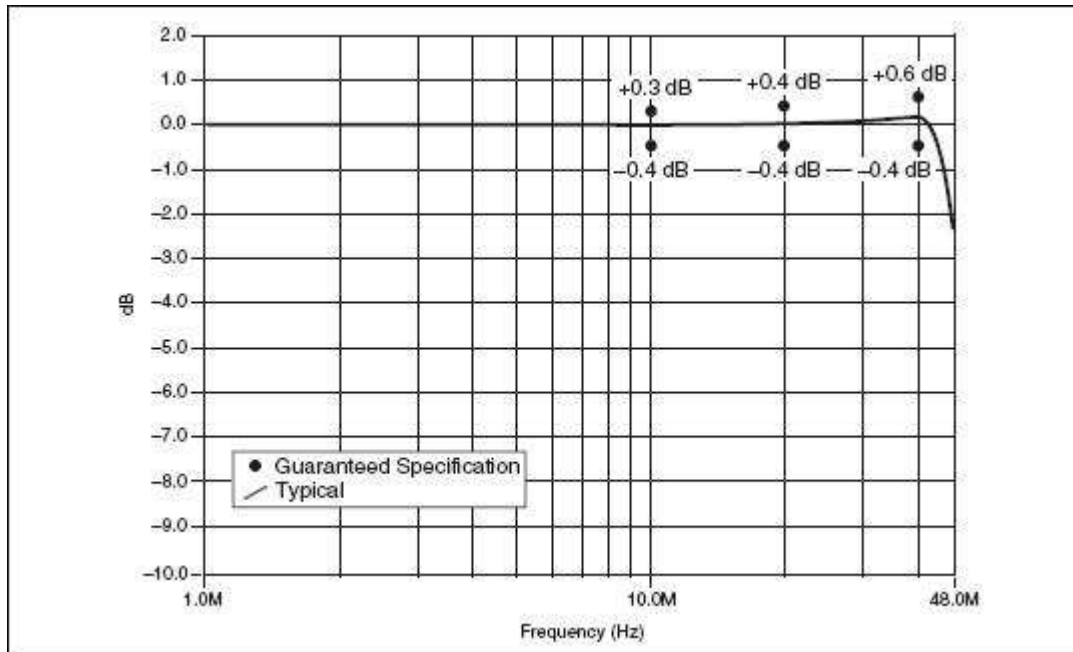


Figura 4.13 Respuesta en frecuencia

Como ilustra el gráfico anterior, la señal de salida es típicamente atenuada -3dB aproximadamente a 43 MHz, que es el ancho de banda especificado para el generador de funciones, en este caso. Del gráfico, entonces, se puede ver que la señal generador funciones, puede emitir señales de forma fiable con un rango de frecuencia entre 0 y 40 MHz.

Un generador de señal especifica los límites dentro de los cuales la amplitud de una señal varía tantos decibelios en función de una gama de frecuencias. Por supuesto, idealmente este valor debería de ser cero decibelios en toda la gama de frecuencias, pero realmente no es así.

El SFDR



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

Es otra especificación importante para caracterizar el comportamiento dinámico de un generador de señal. SFDR especifica la relación entre la amplitud de la frecuencia fundamental que se genera y la amplitud de la frecuencia armónica más destacada. Idealmente, el dominio de la frecuencia de una señal analógica pura tiene toda su potencia concentrada en la frecuencia deseada. Sin embargo, debido al ruido y la no linealidad de los componentes, incluso los mejores generadores de señal también generan contenido de señal en los armónicos (o múltiplos) del tono deseado.

Por ejemplo, cuando se genera una onda senoidal de 10 MHz, se puede observar en los armónicos de 20 MHz, 30 MHz, y así sucesivamente. Estos armónicos son también conocidos como espurios.

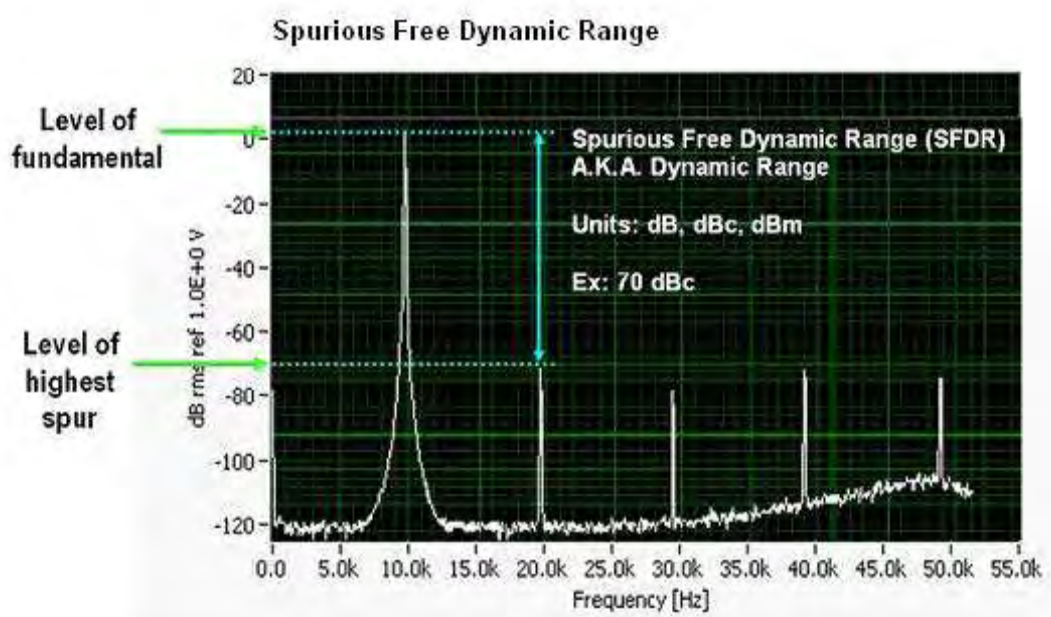


Figura 4.13 Medida del SFDR: el segundo armónico es el tono más alto y el SFDR es de aproximadamente 70 dB, que significa que la amplitud de los tonos armónicos siempre es inferior a .00000001 veces la amplitud de la frecuencia fundamental.

Para medir la SFDR de un generador de señal, se genera un tono una frecuencia dada. Se usa entonces un analizador de espectro para medir la amplitud del tono fundamental y la amplitud de la el tono siguiente más alto. Normalmente, es uno de los armónicos contiguos.

En decibelios, para calcular SFDR se usa la siguiente fórmula:



$$\text{SFDR} = \text{Amplitude of Fundamental (dB)} - \text{Amplitude of Largest Spur (dB)}$$

O también en voltios RMS:

$$\text{SFDR} = 20 \times \log\left(\frac{\text{Amplitude of Fundamental (RMS)}}{\text{Amplitude of Largest Spur (RMS)}}\right)$$

### La distorsión armónica (THD)

Es otra especificación utilizada para caracterizar a los armónicos de las señales analógicas. Sin embargo, a diferencia del SFDR, el THD se utiliza para cuantificar la distorsión de todos los armónicos, no sólo el más alto. Por lo tanto para las señales con un gran número de importantes armónicos, la distorsión armónica total es una especificación que se puede utilizar para representar adecuadamente el efecto que todos estos tienen en el tono fundamental. Como regla general, una señal se muestra visiblemente distorsionada cuando la distorsión armónica total se aproxima a -30 dB.

Se puede calcular mediante la suma la energía de la señal en cada uno de los armónicos dividido por el total de energía de señal en el tono fundamental. Por lo tanto, la ecuación para THD es:

$$\text{THD} = \frac{\sqrt{A_2^2 + A_3^2 + \dots + A_n^2}}{A_{\text{fundamental}}}$$

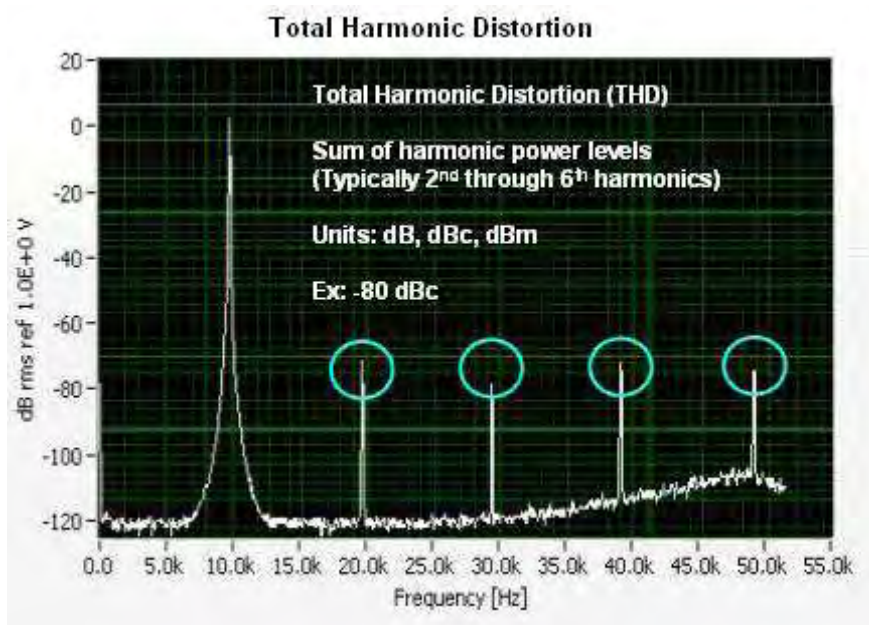


Figura 4.13 Medida del THD

Al elegir un generador de señal, hay que tener en cuenta que el THD generalmente deteriora la señal generada conforme aumenta en frecuencia.

La relación señal-ruido y distorsión (SINAD) es una especificación menos utilizada comúnmente para caracterizar el rendimiento de un generador de señal dinámico. Se define como la relación entre la señal a la potencia total recibida y la señal a la potencia de ruido más distorsión. Generalmente se expresa en decibelios. La relación SINAD se puede calcular con la siguiente fórmula:





En la ecuación anterior, la amplitud de cada variable es una suma de Voltios RMS de todas las componentes espectrales de la señal generada. Como es de esperar, cuanto mayor es la relación SINAD, mejor será la calidad de la señal de salida.

### Número efectivo de bits (ENOB)

Los convertidores digital a analógico deben superar desafíos tales como el ruido del sistema y la no linealidad de los componentes analógicos. Como resultado, una forma de representar la resolución de un determinado generador de señal es una especificación llamada número efectivo de bits (ENOB). Calcularse directamente a partir de la especificación SINAD, el ENOB mide el funcionamiento del DAC del generador de señal entre la fuente de señal generada y las fuentes de ruido y no linealidad.

La manera más fácil de medir ENOB es utilizar la especificación SINAD, medida en decibelios. Por lo tanto, la ecuación para ENOB es como sigue:

$$\text{ENOB} = \left( \frac{\text{SINAD} - 1.76}{6.02} \right)$$

Como un ejemplo, un generador que tiene una SINAD de 64 dB cuando se genera una senoide 1 Mhz. Así, el número efectivo de bits para esta señal es:

$$\text{ENOB} = \left( \frac{64 - 1.76}{6.02} \right) = 10.3 \text{ bits}$$

### Distorsión de intermodulación (IMD)





Es la modulación de amplitud de las señales que contienen dos o más frecuencias diferentes en un sistema con no linealidades. La intermodulación entre cada componente de frecuencia forman señales adicionales en las frecuencias que no son sólo frecuencias armónicas, sino también en la suma y diferencia de las frecuencias originales y en múltiplos de esas frecuencias suma y diferencia.

$$\text{IMD} = \left( \frac{\text{sum of distortion} + \text{difference of distortion}}{\text{Overall Signal Level}} \right)$$

Esta especificación es más importante para banda de base I / Q o de la frecuencia intermedia (IF) generación. Para estas aplicaciones, una señal de mensaje está típicamente superpuesta sobre un portador, por lo que la señal resultante contiene el contenido de frecuencia en múltiples frecuencias.

### La densidad promedia de ruido

Es una especificación que caracteriza el nivel de ruido en la señal generada. También se puede pensar en él como el ruido de la señal. Esta especificación se ve afectada por un número de factores, incluyendo la resolución de bits del DAC y el ruido introducido por los amplificadores. Por lo tanto, se puede caracterizar el ruido introducido en la señal, ya sea con un nivel de ruido o la especificación de densidad promedia de ruido. La siguiente figura muestra el dominio de la frecuencia de una señal que se ha generado con un generador de forma de onda arbitraria.

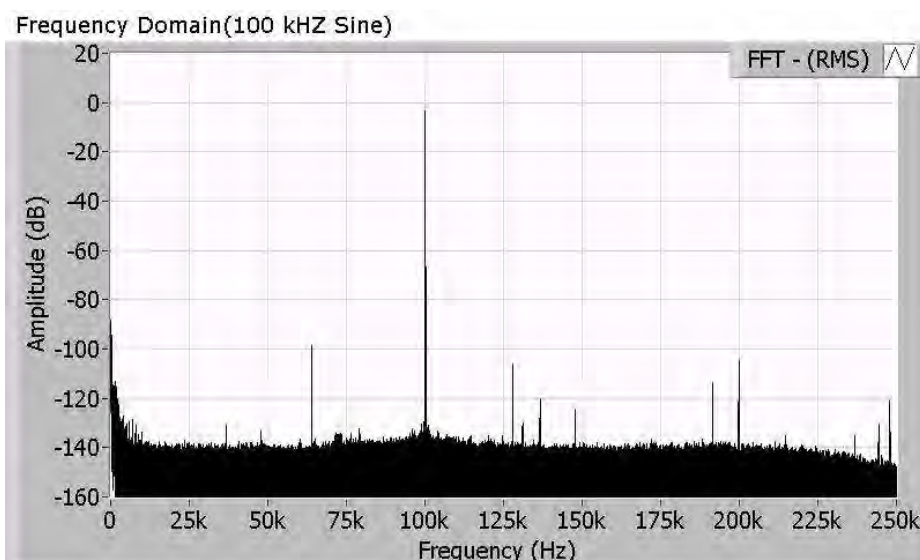


Figura 4.14 Densidad media de ruido de una señal

Como muestra la figura, el ruido de fondo es muy por debajo de  $-130 \text{ dBm / Hz}$ . Además, se puede ver el SFDR es de aproximadamente  $90 \text{ dBc}$ . Se observa en este ejemplo que se está generando un tono de  $100 \text{ kHz}$ .

### Análisis de Fourier

A menudo, se percibe que una señal no es un simple seno o coseno, sino que se parece más a la "suma" entre varias ondas seno y/o coseno. Sin embargo, el teorema de Fourier afirma que cualquier forma de onda en el dominio del tiempo (es decir, cualquiera que se pueda ver en un osciloscopio) puede ser representada por la suma ponderada de senos y cosenos. Esta "suma" de forma de onda por debajo en realidad está compuesta de seno individual y ondas del coseno de frecuencia variable. El mismo "suma" de forma de onda aparece en el dominio de la frecuencia como los valores de amplitud y fase en cada componente de frecuencia (es decir,  $f_0$ ,  $2f_0$ ,  $3f_0$ ).

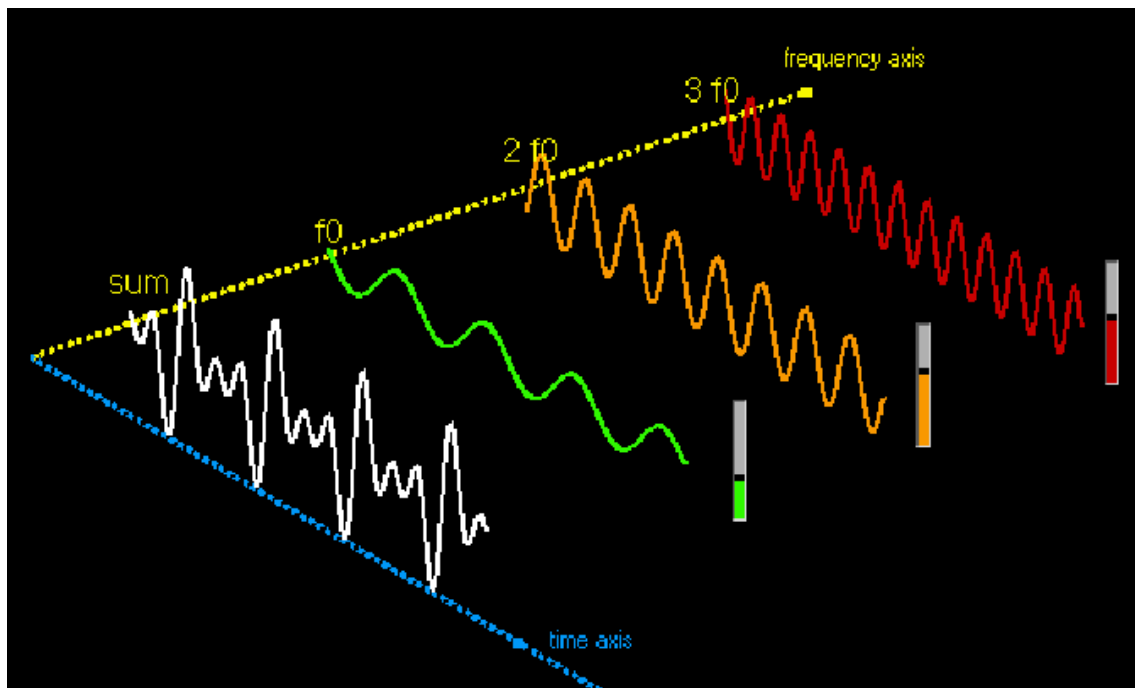


Figura 4.14 Dominio del tiempo vs dominio en frecuencia

La transformada de Fourier convierte una representación dominio del tiempo de una señal en una representación en el dominio de frecuencia. Sin embargo, un ADC muestrea la forma de onda y la transforma en valores discretos. Debido a esta transformación, la transformada de Fourier no funcionará con estos datos discretos. En su lugar, se utiliza la transformada discreta de Fourier (DFT), lo que produce como resultado componentes de dominio en frecuencia en valores discretos. La Transformada Rápida de Fourier (FFT) es una implementación optimizada de un DFT, por tanto lleva menos cálculo que una transformada de Fourier normal.

La transformada de Fourier se define por la ecuación siguiente:



$$X(f) = F\{x(t)\} = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt ,$$

Donde  $x(t)$  es la señal de dominio de tiempo,  $X(f)$  es la FFT, y  $f$  es la frecuencia de analizar.  
Del mismo modo, la DFT viene dada por la ecuación siguiente:

$$X_k = \sum_{i=0}^{n-1} x_i e^{-j2\pi ik/n} \quad \text{for } k = 0, 1, 2, \dots, n-1 ,$$

, donde  $x$  es la secuencia de entrada,  $X$  es la DFT, y  $n$  es el número de muestras, tanto en el tiempo discreto y los dominios discretos de frecuencia.

Para una aplicación directa de la DFT, como se muestra en la ecuación anterior, se requiere aproximadamente  $n^2$  operaciones complejas. Sin embargo, los algoritmos computacionalmente eficientes pueden requerir tan solo  $n \log_2(n)$  operaciones. Estos algoritmos son FFTs.

Cuando se utiliza la DFT, la transformada de Fourier de cualquier secuencia  $x$ , ya sea real o compleja, siempre resulta en una radiografía de la producción compleja secuencia de la siguiente forma:



Una característica inherente DFT es la siguiente:

$$X_{n-i} = X_{-i},$$

En consecuencia,

$$\text{Re}\{X_i\} = \text{Re}\{X_{n-i}\},$$

y además:

$$\text{Im}\{X_i\} = -\text{Im}\{X_{n-i}\}.$$

Estas propiedades simétricas de Fourier de secuencias reales se conocen como conjugados simétricos.

El uso de la FFT para el análisis de frecuencia implica dos relaciones importantes.

La primera relación que enlaza la frecuencia más alta que se puede analizar ( $F_{\max}$ ) a la frecuencia de muestreo ( $f_s$ ).

$$F_{\max} = \frac{f_s}{2}$$

Los enlaces segunda relación de la resolución de frecuencia ( $f$ ) el tiempo de adquisición total ( $T$ ), que está relacionada con la frecuencia de muestreo ( $f_s$ ) y el tamaño de bloque de la FFT ( $N$ ).

$$\Delta f = \frac{1}{T} = \frac{f_s}{N}$$

El espectro de salida FFT es complejo, es decir, cada componente en frecuencia tiene una



magnitud y fase. Las aplicaciones que utilizan el análisis de FFT son bastante extensas. Aquellas aplicaciones que requieren tiempos de respuesta rápidos, tales como el análisis de transitorios, las vibraciones y las pruebas de choque utilizan el análisis de FFT.

### **Ventanas: Optimización del uso de la ventana FFT, es decir, utilización de funciones de la ventana para mejorar las mediciones de frecuencia.**

#### Definición

Las ventanas son funciones matemáticas usadas con frecuencia en el análisis y el procesamiento de señales para evitar las discontinuidades al principio y al final de los bloques analizados.

En procesamiento de señales, una ventana se utiliza cuando nos interesa una señal de longitud voluntariamente limitada. En efecto, una señal real tiene que ser de tiempo finito; además, un cálculo sólo es posible a partir de un número finito de puntos. Para observar una señal en un tiempo finito, la multiplicamos por una función ventana.

Mediante el uso de funciones de la ventana, el resultado de la resolución espectral de su dominio en frecuencia aumentará.

La FFT se comporta como si fuera un banco de filtros de banda estrecha seguido por sus correspondientes bancos de detectores que calculan la suma vectorial de toda la señal mediante el cálculo individual de cada componente de cada filtro paso banda estrecha por el que la señal pasa. En el caso del espectro de potencia, el resultado es un escalar en lugar de datos vectoriales, y el detector sólo indica la energía total que pasa por cada filtro. De nuevo, debido a la duración finita de los datos de origen, los filtros tienden a no ser especialmente selectivos.

Como la siguiente figura, mientras que el lóbulo principal del filtro es razonablemente estrecho, la respuesta en banda lateral del filtro se extiende hacia fuera inconvenientemente.

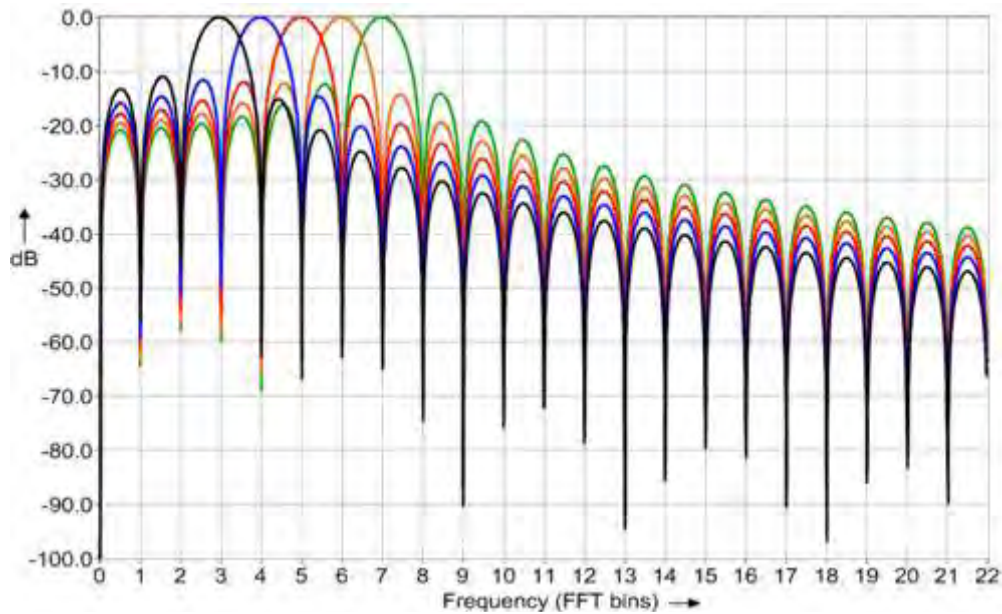


Figura 4.15 FFT. Banco de filtros

Cuando se utiliza la FFT para medir el contenido en frecuencia de los datos, se tendrá que basar el análisis en un conjunto finito de datos. La transformada FFT se supone que el conjunto finito de datos es un período de una señal periódica.

### Número entero de períodos

Cuando la señal es periódica en el intervalo de tiempo de adquisición se recogen un número entero de periodos de señal, la FFT resulta ideal:



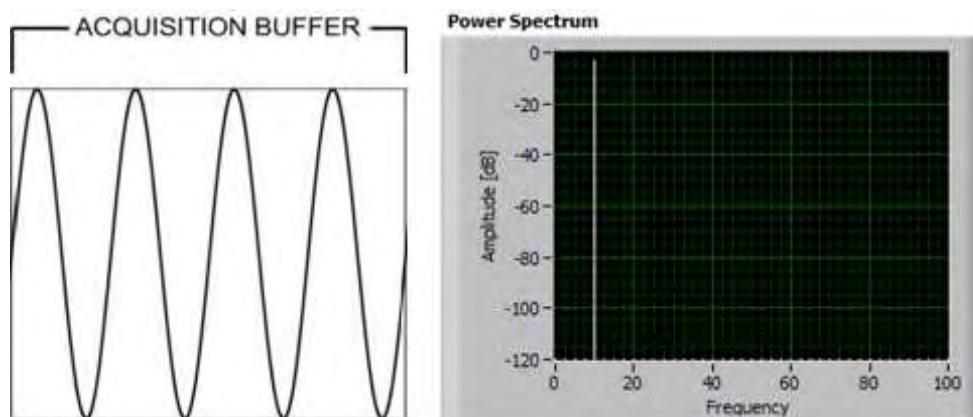


Figura 4.16 Medición de un número entero de períodos resultan FFT ideales

Pero cuando el número de períodos en que la adquisición no es un número entero, los extremos son discontinuos. Como resultado se observan lóbulos laterales indeseados en la gráfica del espectro.

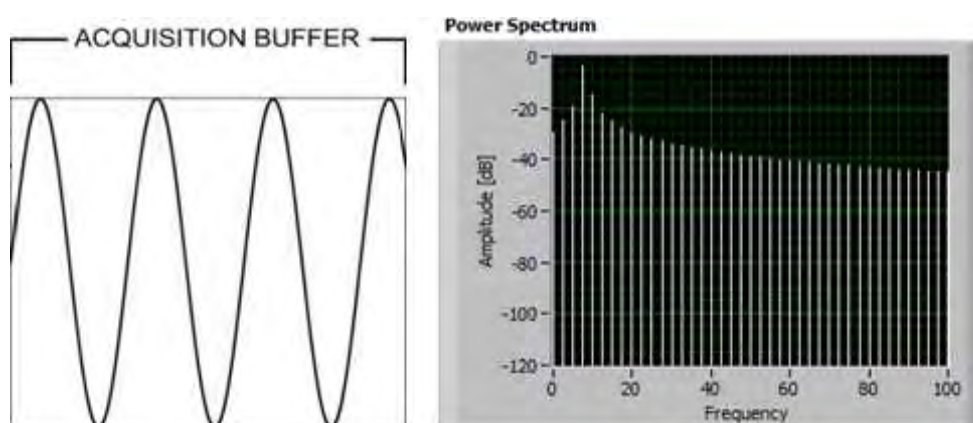


Figura 4.17 La medición de un número no entero de períodos implican lóbulos laterales indeseados en la FFT



Aplicando una función de ventana a la adquisición (número no entero de periodos) antes de calcular la FFT, dará lugar a una mejor resolución espectral.

A esta técnica se le llama aplicar una función de ventana o simplemente aplicar una ventana.

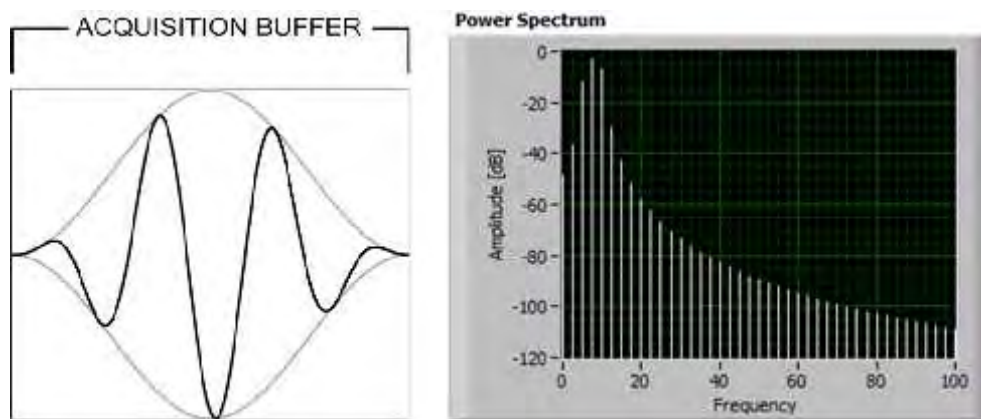


Figura 4.18 Aplicando una ventana minimiza el efecto de fuga espectral

Con cada una de las diferentes funciones de ventana, se dispone de ventajas e inconvenientes dependiendo de la aplicación.

### Ejemplo de una ventana

Aplicando una ventana Blackman-Harris a cuatro periodos de señal antes de realizar la transformada de Fourier FFT reduce en gran medida los lóbulos laterales no deseados en la respuesta en frecuencia, pero a cambio el lóbulo central aumenta. Si esta desventaja no influyera significativamente en la aplicación, sería una buena opción usar esta ventana para



eliminar los lóbulos laterales en la respuesta en frecuencia.

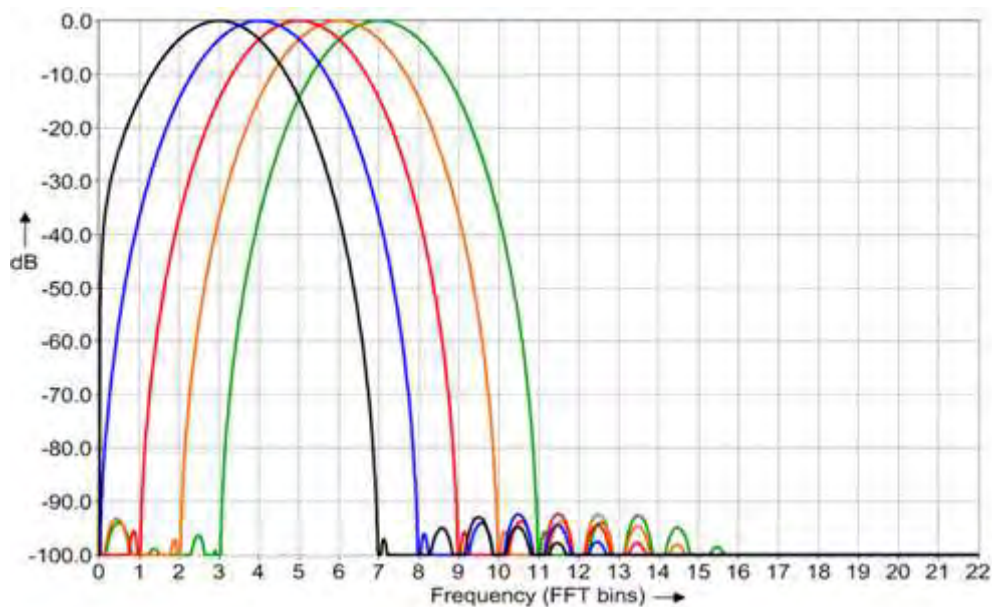


Figura 4.19 Una ventana Blackman-Harris reduce los lóbulos laterales en el dominio de la frecuencia

La siguiente figura muestra el diagrama en dominio de tiempo de algunas de las ventanas comunes. La mayoría de las ventanas comienzan en cero y poseen una amplitud de la unidad en el centro.

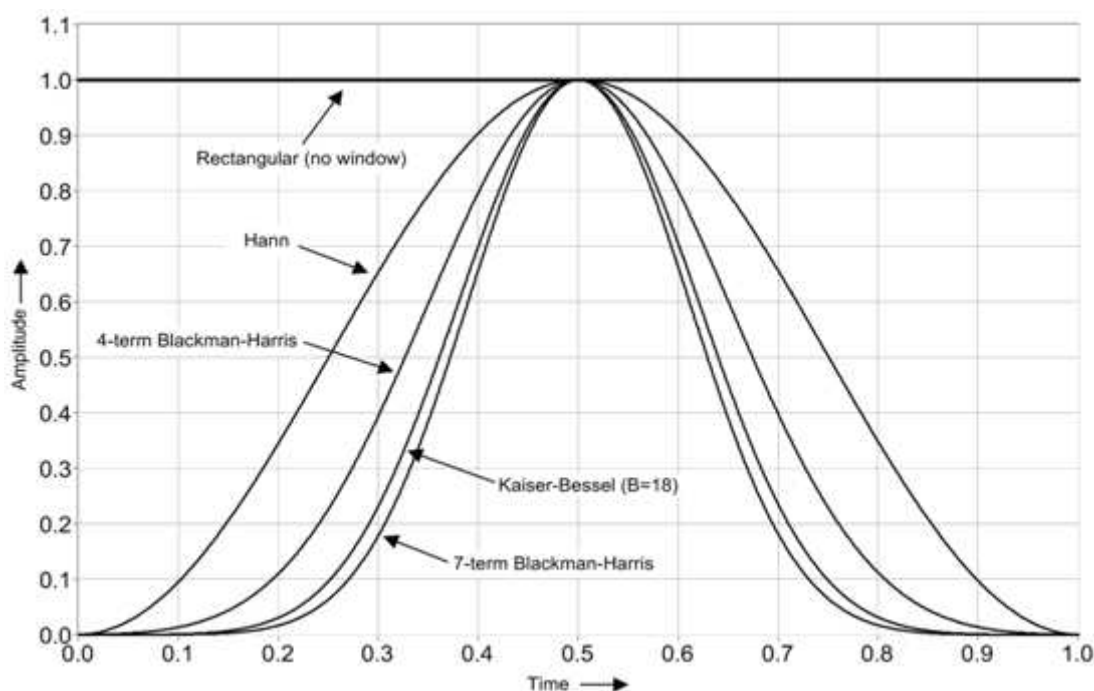


Figura 4.20 Funciones en el dominio temporal de algunas ventanas.

Debido a los lóbulos laterales no deseados en la respuesta en frecuencia, utilizar la FFT con una función de ventana rectangular (o ninguna función de ventana) no es recomendable.

La función de ventana Hanning la función de ventana es útil para mediciones de ruido donde se desea una mejor resolución de la frecuencia, donde los lóbulos secundarios no deseados son moderados y no presentan un problema significativo.

La función de ventana Blackman-Harris es una buena ventana de propósito general, con un rechazo a los lóbulos laterales no deseados del 90 dB, y tiene un lóbulo principal moderadamente ancho.

La función de ventana Kaiser-Bessel tiene un parámetro variable, beta, que hace que la función de ventana sea ajustable. Este parámetro, beta, controla de forma explícita el compromiso existente entre el lóbulo principal y los lóbulos laterales.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

Estás son varios ejemplos de ventanas comunes, pero hay muchas más que se usan de manera cotidiana para diseñar filtros, en automatización, etc.

Potencias de computación futuras, favorecerán el uso de ventanas con mayor rango dinámico.

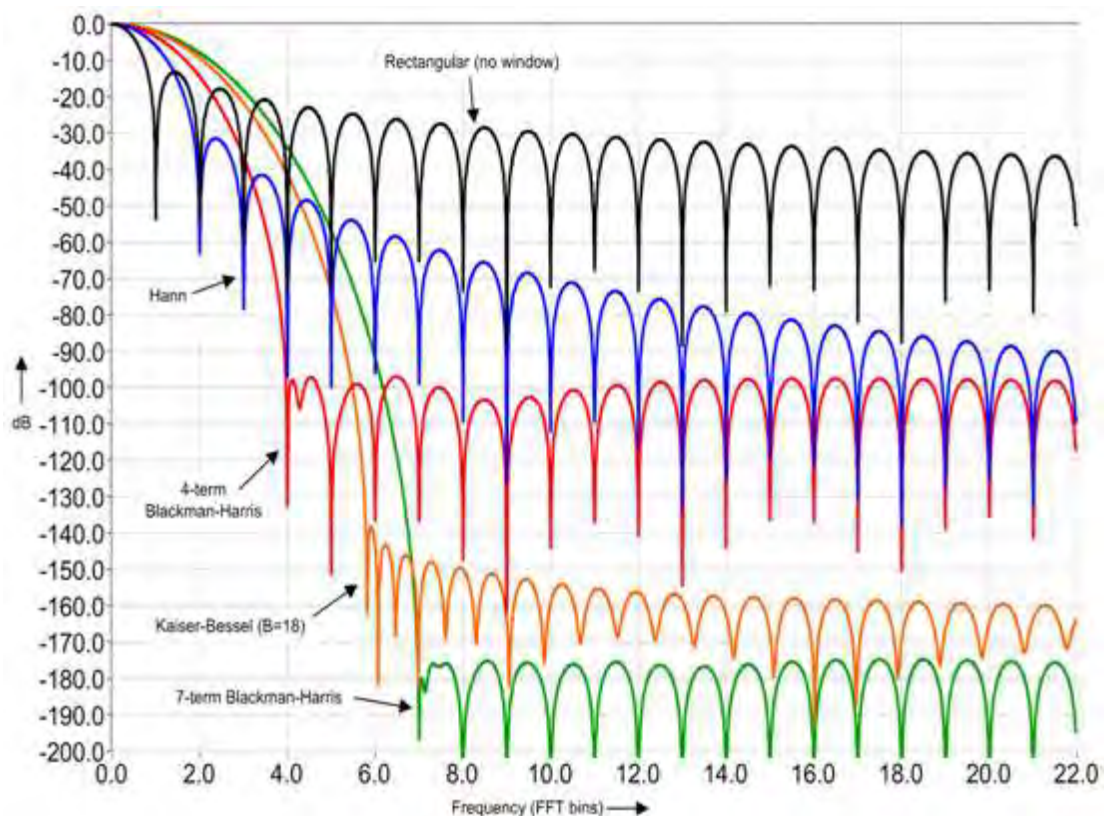


Figura 4.21 La respuesta de frecuencia-dominio de las ventanas

La selección de una función de ventana no es una tarea sencilla. Cada función de la ventana tiene sus propias características y su idoneidad para diferentes aplicaciones. Para elegir una función de ventana, se debe estimar el contenido en frecuencia de la señal, es decir, su respuesta en frecuencia.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

Si la precisión de la amplitud de una componente de frecuencia única es más importante que la ubicación exacta de la componente en una bandeja de frecuencia dada, elegir una ventana con un lóbulo principal de ancho.

Si el espectro de la señal es más bien plana o de banda ancha en el contenido de frecuencia, utilizar la ventana de uniforme, o ninguna ventana.

En general, el uso de la función de ventana Hanning es satisfactorio en el 95% de los casos, ya que tiene buena resolución de frecuencia y la fuga espectral es reducida.





## **Capítulo 5: Ingeniería del Hardware**

### **Introducción**

National Instruments proporciona cientos de dispositivos de adquisición de datos para elegir sobre una gran variedad de buses, por lo que puede ser complicado elegir de entre tantos dispositivos el que mejor conviene para nuestra aplicación concreta. Cada bus posee ventajas diferentes para optimizar el rendimiento, latencia, portabilidad, o la distancia que hay al sensor de medida.

En el caso de este proyecto concreto, por disponibilidad se ha elegido la tarjeta de adquisición DAQPad 6020-E, la cual posee un bus USB por el cual se conecta al PC.

### **Interfaz hardware**



Figura 5.1 Tarjeta DAQPad 6020-E

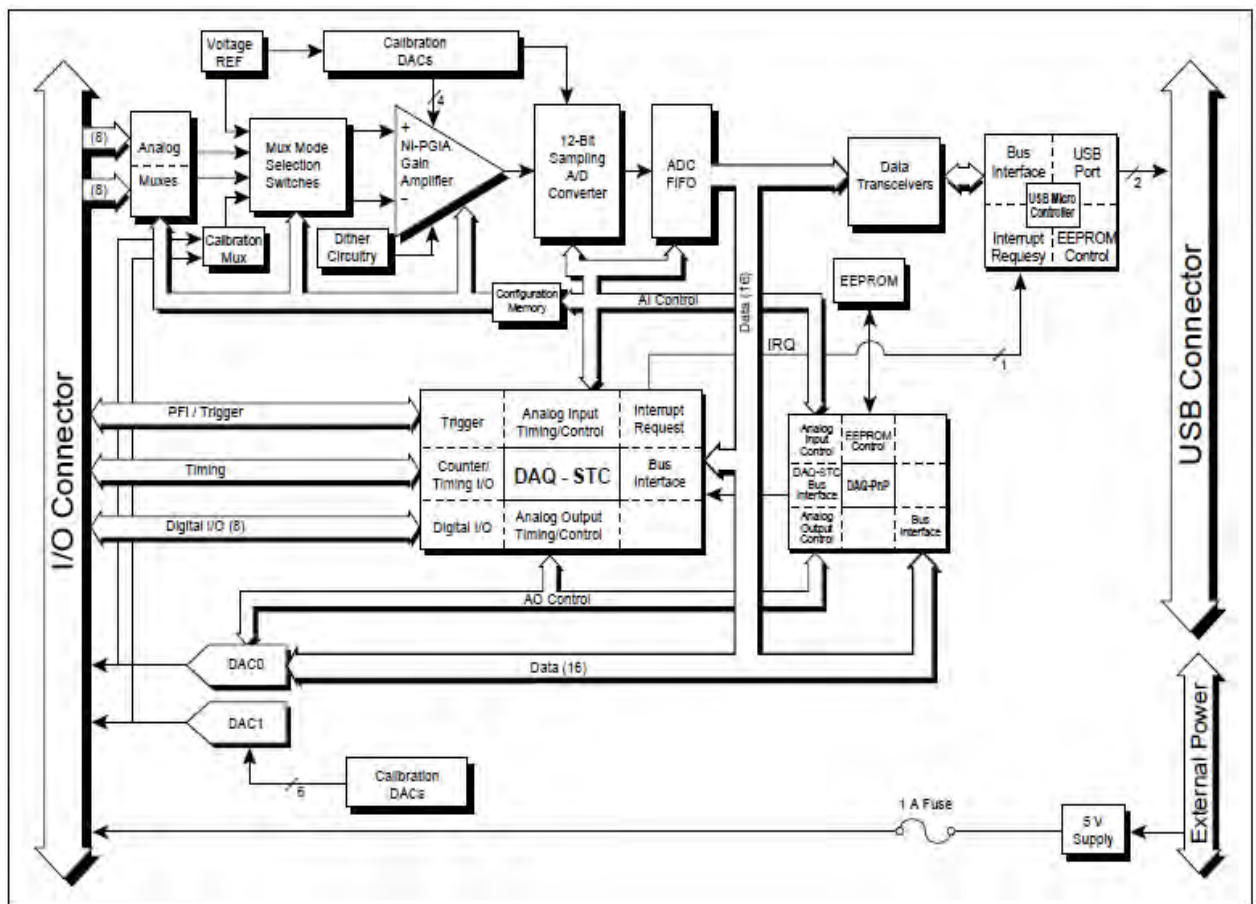




# Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

La interfaz hardware que se va a utilizar es una tarjeta de adquisición de datos de National Instruments, en particular la DAQPad-6020E

## Hardware Overview de la tarjeta DAQPad-6020E



Modos de configuración de entrada:



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

La tarjeta de adquisición DAQPad-6020E posee tres modos de entrada de señal analógica, los cuales son configurables por software. Estos modos son:

-Entrada single-ended no referenciada a masa (NRSE)

-Entrada single-ended referenciada a masa (RSE)

-Entrada diferencial (DIFF)

NRSE: Un canal configurado en el modo NRSE utiliza una línea de canal analógico, la cual se conecta a la entrada positiva del PGIA. La entrada negativa del PGIA se conectará a la entrada analógica AISENSE.

RSE: Un canal configurado en el modo RSE también utiliza una línea de canal analógico, la cual se conectará a la entrada positiva del PGIA. En cambio la entrada negativa del PGIA se conectará a la entrada AIGND, que es la masa analógica.

DIFF: Un canal configurado en el modo DIFF también utiliza dos líneas de canal analógico. Una se conectará a la entrada positiva del PGIA, y la otra línea se conectará a la entrada negativa del PGIA.



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

Input	Signal Source Type	
	Floating Signal Source (Not Connected to Building Ground)	Grounded Signal Source
	Examples <ul style="list-style-type: none"> <li>• Ungrounded Thermocouples</li> <li>• Signal conditioning with isolated outputs</li> <li>• Battery devices</li> </ul>	Examples <ul style="list-style-type: none"> <li>• Plug-in instruments with nonisolated outputs</li> </ul>
Differential (DIFF)	<p>See text for information on bias resistors.</p>	
Single-Ended — Ground Referenced (RSE)		<p><b>NOT RECOMMENDED</b></p> <p>Ground-loop losses, <math>V_g</math>, are added to measured signal</p>
Single-Ended — Nonreferenced (NRSE)	<p>See text for information on bias resistors.</p>	



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

### Universidad Politécnica de Madrid

#### **Polaridad y rangos de entrada:**

La tarjeta de adquisición DAQPad-6020E posee dos polaridades de entrada, la unipolar y la bipolar. Si configuramos una entrada unipolar, el rango de tensión de entrada se encontrará entre cero y  $V_{ref}$ , siendo  $V_{ref}$  la tensión de referencia positiva (10V).

Una entrada bipolar en cambio tendrá un rango de tensión de entrada entre  $-V_{ref}/2$  y  $+V_{ref}/2$ , lo cual significa que el rango de tensión de entrada se encontrará entre **+5V**.

La ganancia programable por software incrementa la flexibilidad para poder convertir las señales analógicas a señales digitales con los ADC de la tarjeta, de manera que podremos introducir más precisión en la medida a mayor ganancia:



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

Range Configuration	Gain	Actual Input Range	Precision <sup>1</sup>
0 to +10 V	1.0	0 to +10 V	2.44 mV
	2.0	0 to +5 V	1.22 mV
	5.0	0 to +2 V	488.28 $\mu$ V
	0.0	0 to +1 V	244.14 $\mu$ V
	0.0	0 to +500 mV	122.07 $\mu$ V
	50.0	0 to +200 mV	48.83 $\mu$ V
	100.0	0 to +100 mV	24.41 $\mu$ V
-5 to +5 V	0.5	-10 to +10 V	4.88 mV
	1.0	-5 to +5 V	2.44 mV
	2.0	-2.5 to +2.5 V	1.22 mV
	5.0	-1 to +1 V	488.28 $\mu$ V
	10.0	-500 to +500 mV	244.14 $\mu$ V
	20.0		122.07 $\mu$ V
	50.0	-250 to +250 mV	48.83 $\mu$ V
	100.0		24.41 $\mu$ V
		-100 to +100 mV	
		-50 to +50 mV	

La polaridad de la entrada y el rango que seleccionamos depende del rango de la señal de entrada. Un rango de entrada grande implica una gran variación en la señal, por lo que la resolución de tensión en la medida disminuirá. Por el contrario, con señales con rango de la señal de entrada más pequeño implicaría variaciones en la señal más pequeñas, por lo que la resolución de tensión en la medida aumenta.

**Dither:**

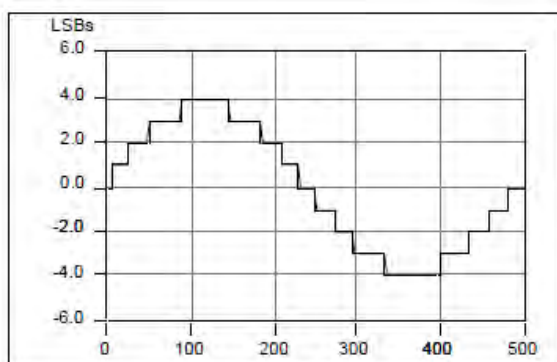


## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

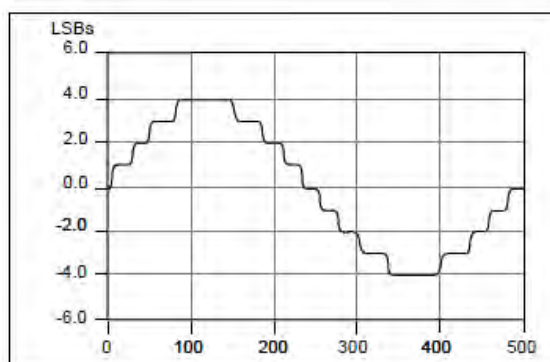
Universidad Politécnica de Madrid

Para incrementar aún más la resolución en la medida, la tarjeta posee la opción de añadir 0.5 LSB Vrms de ruido blanco Gaussiano a la señal que le entra al convertidor analógico-digital. Añadir este ruido implica un promedio de aumento de resolución para calibrar la medida y para el análisis espectral.

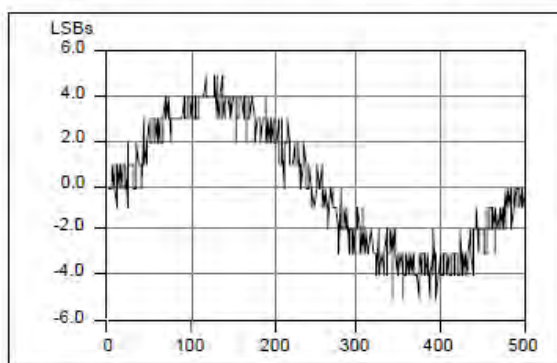
Para medidas de señales en continua, por ejemplo cuando se comprueba la calibración de un dispositivo, añadir este ruido elimina los efectos de cuantificación y reduce el ruido en la medida, resultando la resolución incrementada. Esta opción también es configurable por software. A continuación se presenta el efecto del Dither en la adquisición de pequeña señal ( $\pm 4$  LSB):



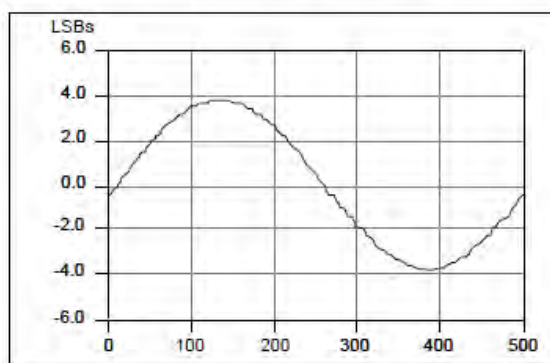
a. Dither Disabled; No Averaging



b. Dither Disabled; Average of 50 Acquisitions



c. Dither Enabled; No Averaging



d. Dither Enabled; Average of 50 Acquisitions



**Entradas/Salidas digitales:**

La tarjeta de adquisición DAQPad-6020E posee ocho líneas de entrada/salida de propósito general. Se puede configurar cada línea de forma individual por software para que sean líneas de entrada o salida.

**Asignamiento de pines de entrada salida:**

El conector de la tarjeta de adquisición DAQPad-6020E tiene 68 pines de entrada/salida:





# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

ACH8	34	68	ACH0
ACH1	33	67	AIGND
AIGND	32	66	ACH9
ACH10	31	65	ACH2
ACH3	30	64	AIGND
AIGND	29	63	ACH11
ACH4	28	62	AISENSE
AIGND	27	61	ACH12
ACH13	26	60	ACH5
ACH6	25	59	AIGND
AIGND	24	58	ACH14
ACH15	23	57	ACH7
DAC0OUT	22	56	AIGND
DAC1OUT	21	55	AOGND
EXTREF	20	54	AOGND
DIO4	19	53	DGND
DGND	18	52	DIO0
DIO1	17	51	DIO5
DIO6	16	50	DGND
DGND	15	49	DIO2
+5 V	14	48	DIO7
DGND	13	47	DIO3
DGND	12	46	SCANCLK
PFI0/TRIG1	11	45	EXTSTROBE*
PFI1/TRIG2	10	44	DGND
DGND	9	43	PFI2/CONVERT*
+5 V	8	42	PFI3/GPCTR1_SOURCE
DGND	7	41	PFI4/GPCTR1_GATE
PFI5/UPDATE*	6	40	GPCTR1_OUT
PFI6/WFTRIG	5	39	DGND
DGND	4	38	PFI7/STARTSCAN
PFI9/GPCTR0_GATE	3	37	PFI8/GPCTR0_SOURCE
GPCTR0_OUT	2	36	DGND
FREQ_OUT	1	35	DGND

**Figure 4-1.** I/O Connector Pin Assignment for the DAQPad-6020E 68-Pin I/O

Utilizaremos mapa de pines para conectar a las entradas las señales a medir y/o analizar.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

### **Consideraciones sobre las señales analógicas en el mapa:**

Las entradas analógicas, en el mapa, se corresponden con las señales ACHx <0..15>, AISENSE, y AIGND. Las entradas ACHx están unidas a 16 canales de entrada de la tarjeta.

En modo single-ended, las entradas ACHx están encaminadas a la entrada positiva del PGIA.

En modo diferencial, las entradas ACH<0..7> están encaminadas a la entrada positiva del PGIA, y las entradas ACH<8..15> están encaminadas a la entrada negativa del PGIA.

En modo NRSE, la entrada AISENSE está conectada internamente a la entrada negativa del PGIA cuando los correspondientes canales están seleccionados.

AIGND es una entrada analógica común que está encaminada directamente al punto de masa de la tarjeta. Por tanto, podemos usar esta entrada como punto de masa común del sistema, si es necesario.

### **Consideraciones sobre el PGIA(Amplificador de instrumentación programable):**

Las conexiones de las señales analógicas de entrada de la tarjeta dependen de la configuración de los canales analógicos que estamos usando y del tipo de la fuente de señal. Con las diferentes configuraciones, podemos usar el PGIA de diferentes modos.

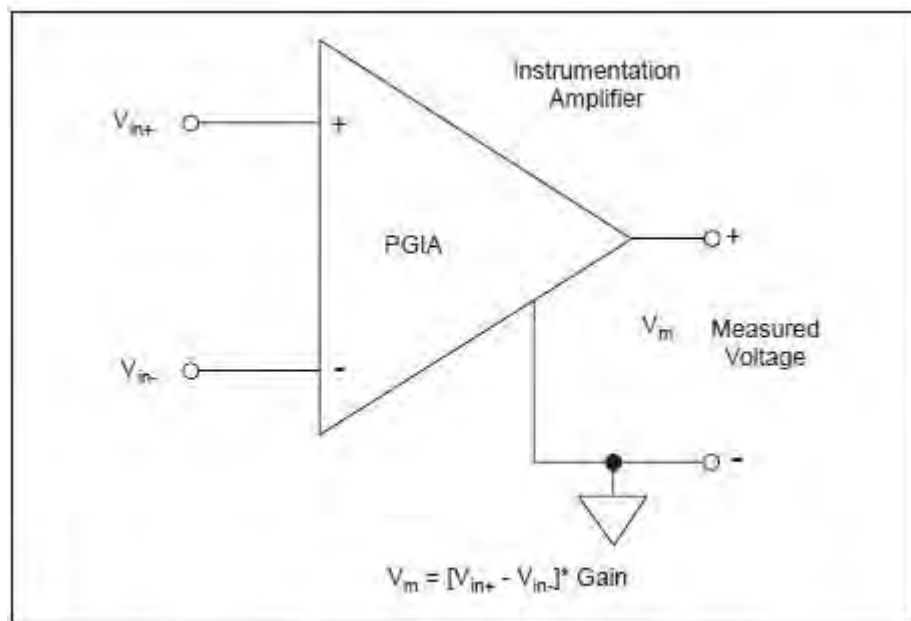


Figure 4-3. DAQPad-6020E PGIA

El PGI aplica una ganancia y un rechazo al modo común, además de una alta impedancia de entrada a las señales analógicas conectadas a la tarjeta.

Además, el PGI está referenciado a la masa analógica de la tarjeta AIGND. Los conversores analógico/digital de la tarjeta están conectados a la salida del PGI.

#### Control de tiempos:

Todo el control externo de las señales temporales que introducimos en la tarjeta DAQPad-6020E está encaminado a través de diez entradas programables, las cuales están nombradas como PFIx <0..9>. Estas entradas son bidireccionales.

Si están configuradas como salidas, no serán programables y reflejarán el estado de la cualquier adquisición de datos que realicemos, generación de formas onda y de las señales de propósito general.



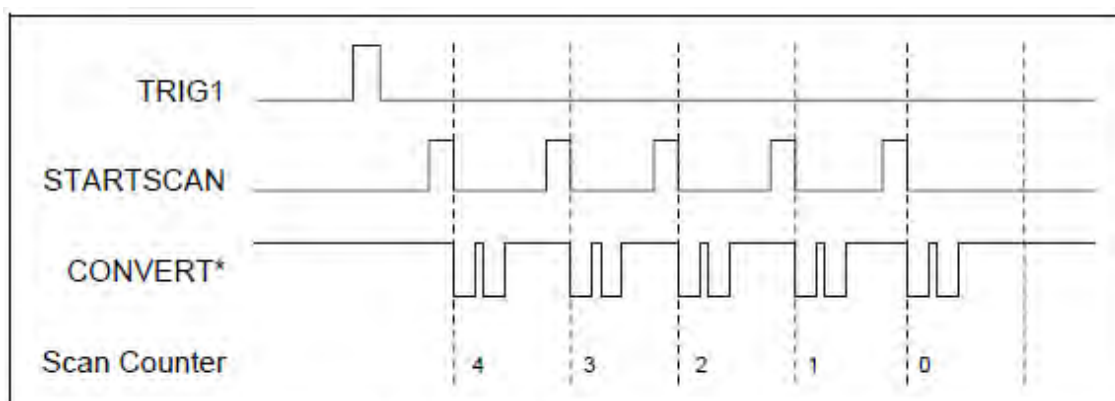
Si están configuradas como entradas, las señales PFI serán programables y podrán controlar la adquisición de datos, la generación de formas de onda y las señales de propósito general que podamos utilizar. Además, podemos configurar cada entrada de forma individual para que detecten tanto flancos de subida/bajada como por nivel, añadiendo selección de polaridad (detección de flanco de subida/bajada). En detección por flanco, el mínimo ancho de pulso requerido es de 10 nanosegundos. No hay máximo ancho de pulso requerido. En detección por nivel, no hay ni ancho de pulso máximo ni mínimo requerido.

Todas las conexiones de entrada/salida de la tarjeta están referenciadas a la masa digital de la tarjeta DGND.

### **Adquisición de datos:**

Las señales encargadas de la adquisición de datos son SCANCLK, EXTSTROBE, TRIG1, TRIG2, STARTSCAN, CONVERT, AIGATE, y SISOURCE.

La adquisición de datos post-disparo permite observar solo la información que se adquiere después de que evento de disparo es recibido. Un ejemplo típico de una adquisición de datos post-disparo se refleja a continuación:



**Figure 4-14. Typical Posttriggered Acquisition**



SCANCLK: Esta señal es únicamente de salida, la cual genera un pulso que ocurre aproximadamente de cincuenta hasta cien nanosegundos después de que la conversión analógico digital empieza. El ancho de pulso de esta señal dura aproximadamente de cuatrocientos a quinientos nanosegundos si está habilitada por software.

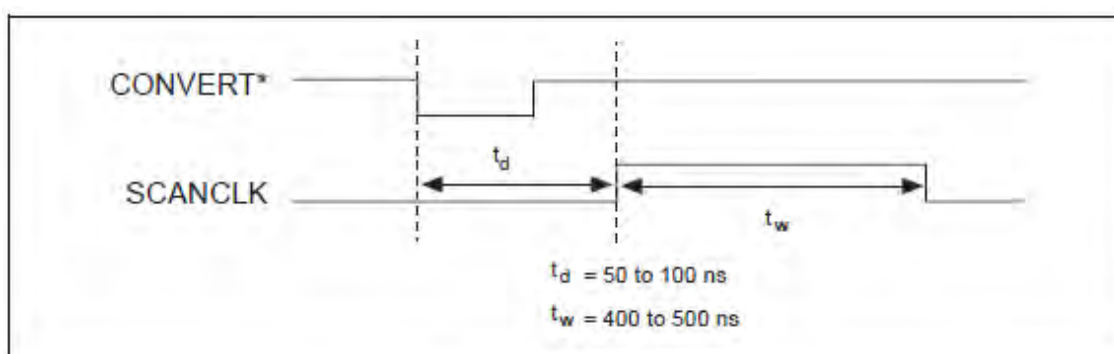


Figure 4-16. SCANCLK Signal Timing

EXTSTROBE: Esta señal también es únicamente de salida, la cual genera o un pulso o una secuencia de ocho pulsos en el modo hardware-strobe. Un dispositivo externo podrá usar esta señal para latched señales o eventos de disparo.

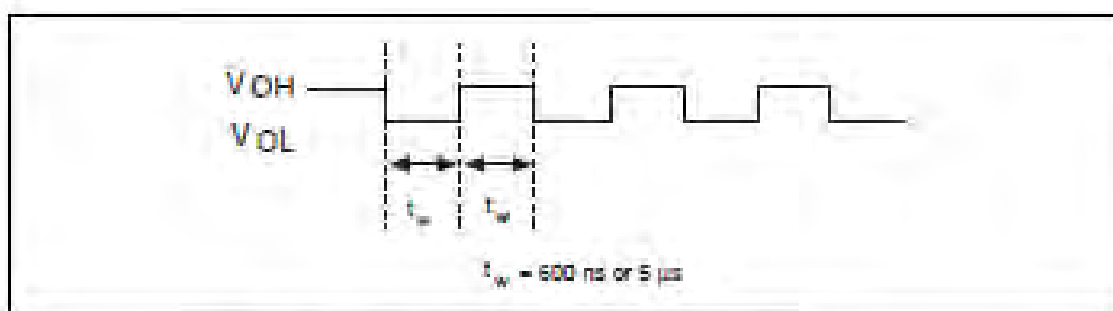


Figure 4-17. EXTSTROBE\* Signal Timing

TRIG1: Cualquier pin PFI de la tarjeta puede encaminar la señal TRIG1, pero está disponible como salida en el pin PFI0/TRIG1. Si está configurada como entrada, la señal TRIG1 está



configurada como detector de flancos de subida o bajada. La detección de flanco comienza con la llegada de los datos a adquirir para ambos tipos de adquisición, la adquisición post-disparo y la adquisición pre-disparo. Como salida, TRIG1 refleja la acción de comienzo de adquisición de la secuencia de datos, solo si la adquisición esta externamente provocada por otra señal PFI. Además, como salida se activa como tri-estado al inicio.

El dispositivo además utiliza la señal TRIG1 para iniciar las operaciones de adquisición de datos pre-disparo. En la mayoría de aplicaciones pre-disparo, se genera un disparo por software. La señal TRIG2 funciona exactamente igual que TRIG1.

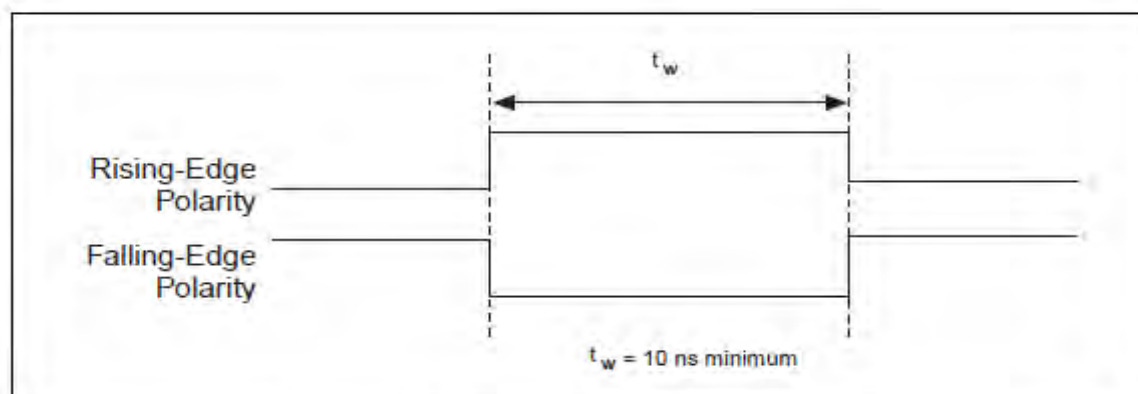


Figure 4-18. TRIG1 Input Signal Timing

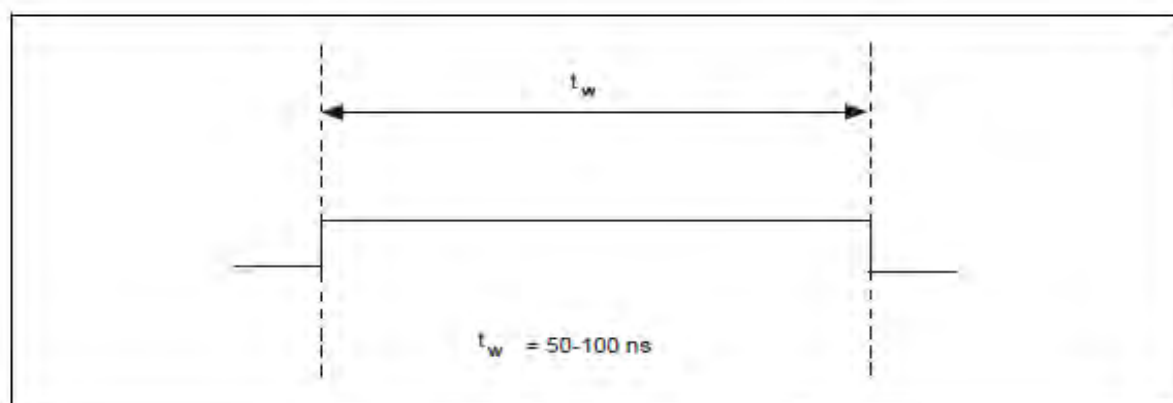


Figure 4-19. TRIG1 Output Signal Timing



STARTSCAN: Cualquier pin PFI de la tarjeta puede encaminar la señal STARTSCAN como señal de entrada, pero está disponible como salida en el pin PFI7/ STARTSCAN.

Si está configurada como entrada, la señal STARTSCAN está configurada como detector de flancos de subida o bajada. La selección del tipo de flanco de STARSCAN hace que se pueda iniciar el escaneo de la señal. El contador de muestras empieza con la señal CONVERT.

Si está configurada como salida, la señal STARTSCAN refleja el pulso de comienzo actual que inicia el escaneo, solo si está externamente provocado por otro PFI.

En este modo, hay dos opciones de salida. La primera es un pulso activo a nivel alto con un ancho de pulso de entre cincuenta a cien nanosegundos, el cual indica el inicio de un escaneo. La segunda, es un pulso también activo a nivel alto, el cual finaliza en el comienzo de la última conversión de la adquisición, por tanto indica que una adquisición está en progreso. En todo caso, al inicio la señal STARTSCAN se encuentra en estado 'tri-state'.

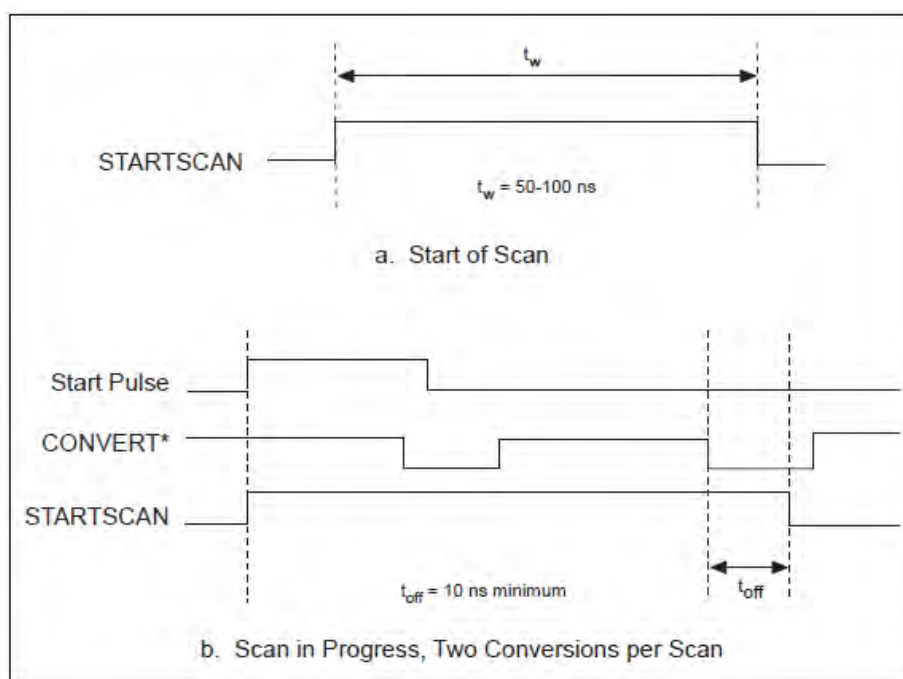


Figure 4-23. STARTSCAN Output Signal Timing



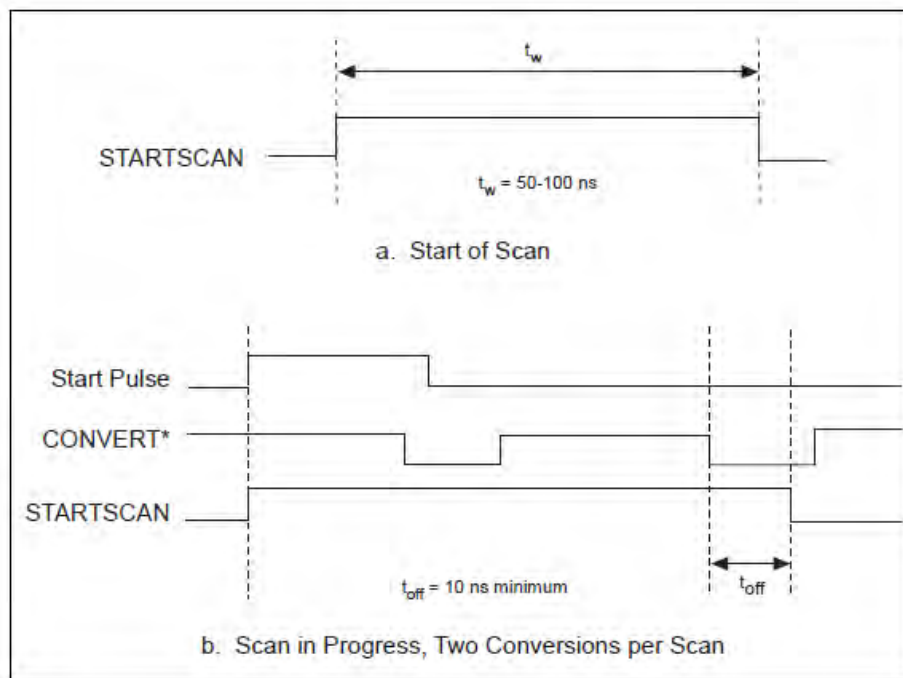


Figure 4-23. STARTSCAN Output Signal Timing

Cabe destacar, que en la tarjeta DAQPad-6020E se encuentra un contador interno que genera la señal STARTSCAN automáticamente, a no ser que se prefiera generar la señal STARTSCAN a través de una fuente externa. Este contador comienza la cuenta cuando se activa la señal TRIG1 y el contador para la cuenta por software o bien puede ser la fuente externa la que deje de generar señales STARTSCAN.

CONVERT: Cualquier pin PFI también puede encaminar la señal de entrada CONVERT, la cual también está disponible como pin de salida en un pin PFI2/CONVERT. En las figuras 4-16 y 4-17 se puede observar la relación entre la señal CONVERT en la secuencia de adquisición de datos.

Como entrada, la señal CONVERT está configurada en modo detector de flancos, pudiendo configurar la polaridad del flanco como flancos de subida y bajada, mediante cualquier pin PFI. Cuando la señal CONVERT detecta un flanco activo, se empieza una conversión analógico a digital.



Como salida, la señal CONVERT refleja el estado actual del pulso que está conectado al conversor analógico/digital. Al inicio, está en estado 'tri-state'.

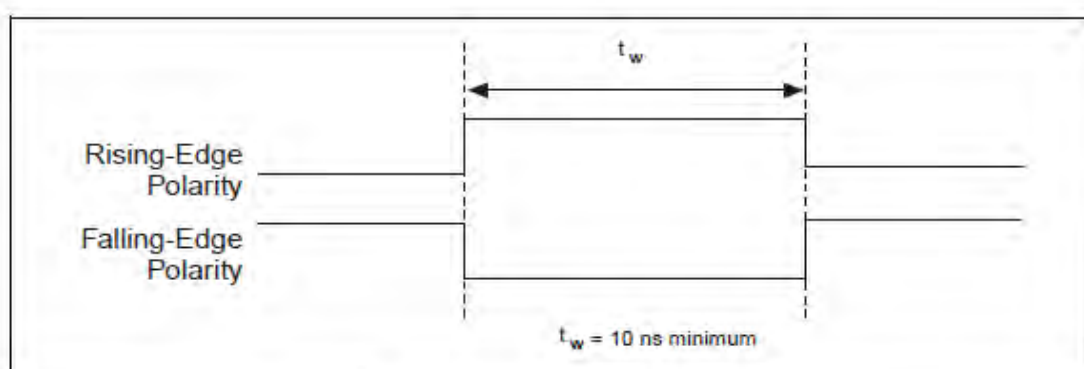


Figure 4-24. CONVERT\* Input Signal Timing

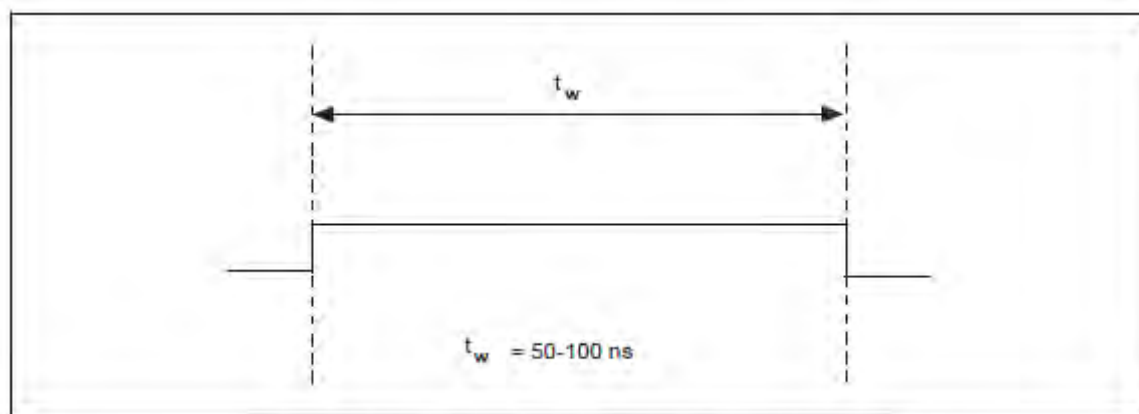


Figure 4-25. CONVERT\* Output Signal Timing

Nota importante: el ADC cambia su modo a retención en aproximadamente unos 60 nanosegundos dentro del flanco activo actual. Este retraso en el modo retención está afectado, o es función de la temperatura, pero no varía entre conversión y conversión.

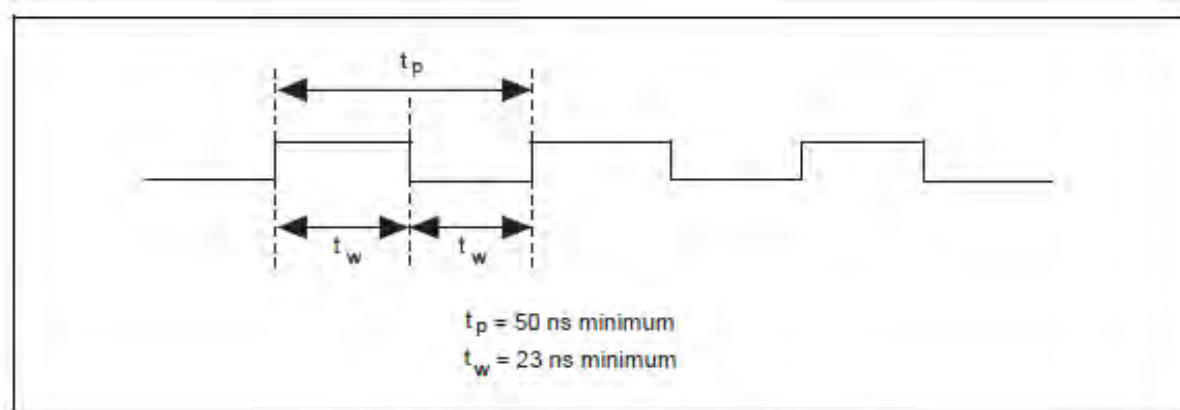
El contador de muestras de adquisición comienza, como ya se ha mencionado, con la señal STARTSCAN y continúa contando, recargándose el mismo hasta que el escaneo ha finalizado.



Una vez que el escaneo de la muestra finaliza, automáticamente se recarga el mismo para esperar al siguiente pulso STARTSCAN.

**AIGATE:** la señal AIGATE se comporta como un 'enable' del conversor analógico digital, de manera que no se encuentra activo, no se pueden producir escaneos. Cabe destacar, de esta señal, que no puede interrumpir una conversión en curso en caso de que se active a mientras una conversión de una muestra se está produciendo en el instante que se activa, pero la siguiente conversión ya no se producirá. La polaridad a la cual se activa esta señal ( nivel alto o bajo ) es configurable exactamente igual que en las otras señales mencionadas anteriormente.

**SISOURCE:** la señal SISOURCE, la cual no está disponible como salida, únicamente como entrada, sirve como reloj del contador generador de señales STARTSCAN, si queremos que un reloj externo controle el contador. La máxima y mínima frecuencia de reloj permitida se encuentra entre 20MHz y 100KHz, y como las demás señales, esta señal puede ser activa a nivel alto o bajo.



**Figure 4-26. SISOURCE Signal Timing**

La tarjeta DAQPad-6020E también hace posible la generación de formas de onda mediante conversores D/A, pero en el presente proyecto solo interesa la adquisición de datos para procesar las señales analógicas adquiridas en el osciloscopio y analizador de espectros virtual, por software, por lo que no se realizará una explicación de cómo la tarjeta realiza la generación de formas de onda mediante DACs.



### **Consideraciones a tomar en cuenta, en relación a la precisión en las medidas**

El ruido en el ambiente puede afectar gravemente a la precisión de las medidas que realiza la tarjeta DAQPad-6020E, si no se toma especial cuidado con el cableado que une la señal que queremos medir y el dispositivo.

Se tomarán una serie de consideraciones para que las medidas sean lo más precisas posibles. Estas medidas las aconseja el fabricante, en el manual de usuario de la tarjeta, pero se pueden aplicar a cualquier encaminamiento entre señales analógicas y dispositivos:

- 1- Configuraremos las entradas analógicas en modo diferencial, para rechazar lo máximo posible el ruido en modo común.
- 2- Usaremos pares de cables protegidos, y además estos pares estarán enrollados entre sí, para conectar la señal analógica al dispositivo. Este tipo de cableado es bueno si la señal atraviesa áreas con grandes campos electromagnéticos, o con grandes interferencias electromagnéticas.
- 3- Los cables deben estar lo más lejos posible de fuentes de ruido. La fuente de ruido más común en este tipo de aplicaciones (tarjeta de adquisición conectada a PC) es el monitor. Por tanto, separaremos lo más posible el monitor de las señales analógicas que mediremos.

Dado que el sistema a realizar es una aplicación de laboratorio, se espera no encontrar fuentes de ruido ambiental o electromagnético tales como líneas de alto voltaje o cualquier otro tipo de sistema, como motores eléctricos, transformadores, etcétera, que puedan inducir corrientes o tensiones adicionales a las señales de la tarjeta, o a la propia tarjeta. Por lo que es suficiente tomar las consideraciones anteriormente expuestas.

### **Calibración**



Por último, hay que discutir los procedimientos de calibración de la tarjeta DAQ. Si usamos el Driver Software de NI-DAQ propio de la tarjeta, se incluyen las funciones de calibración para completar todos los pasos del proceso de calibración.

La calibración se refiere al proceso de minimizar los errores de medida a la salida, realizando pequeños ajustes en la circuitería hardware. Este proceso de calibración es común en la gran mayoría de aplicaciones de medida, ya que todo hardware instrumental o de medida requiere unos ajustes para que la medida sea lo más precisa posible, debido a la no idealidad de los componentes hardware que forman la circuitería. En caso de no realizar la calibración, se podrían tener errores no deseados de offset, ganancia y/o errores de linealidad.

La calibración en esta tarjeta puede hacerse de tres maneras, pero una de ellas no es demasiado precisa, así que se descarta. A continuación se describirán las dos maneras que se seguirán:

**Calibración interna:** la tarjeta DAQPad-6020E puede medir y corregir casi todos los parámetros de calibración internos, sin que haga falta ninguna calibración externa (en principio), gracias a un driver que nos proporciona National Instruments para esta tarjeta en particular. Los errores de calibración se minimizarán hasta ser prácticamente despreciables, pero hay que tener especial atención en el error de ganancia en función de la temperatura. El fabricante, en este caso National Instruments nos recomienda comprobar este error, y en caso de que sea significativo, ayudarnos de la calibración externa.

**Calibración externa:** la calibración externa se realiza mediante una tensión de referencia externa, la cual es conocida y la ajusta el programador. El valor de esta tensión externa de referencia puede también almacenarse en la EPROM de la tarjeta, de manera que para adquisiciones en el mismo entorno de trabajo, solo hará falta calibrar el dispositivo una vez. Cabe destacar, aunque es obvio, que la tensión de referencia externa debe ser lo más exacta posible. El fabricante recomienda que esta tensión de referencia tenga una precisión de al menos 0.005% en valor absoluto.



## **Capítulo 6: Ingeniería del Software**

### **Introducción**

Una vez claros los requerimientos del sistema, habiendo hecho un estudio del estado del arte de los componentes de la aplicación, y un análisis de viabilidad y planificación, se va a pasar a la realización de un diseño preliminar del lo que va a ser el software que va a funcionar junto a la tarjeta de adquisición. Recordemos que la aplicación software constará de:

- Un generador de funciones virtual
- Un analizador de espectro virtual
- Un osciloscopio virtual

Después del diseño preliminar, se mostrará el diseño detallado de cada uno de los módulos, mediante capturas de pantalla del código en LabVIEW, ya depurado. También se mostrarán algunas pruebas realizadas de los módulos actuando de forma independiente. En el próximo capítulo, 'Integración y test', los módulos se unirán, junto a la interfaz hardware, formando la aplicación completa.

**Nota: todos los nombres, etiquetas, notas y comentarios del código, por supuesto estarán en inglés, ya que el código debe ser exportable. Cualquier programador de cualquier parte del mundo debe de poder entender las descripciones y las funcionalidades de la aplicación.**

A continuación se presenta de nuevo el esquema de la planificación del modelo en cascada. Esto servirá de recordatorio sobre los pasos que se están llevando a cabo.



## Modelo Cascada



Figura 6.1 Esquema modelo cascada

### Generador de funciones

El módulo software que compone el generador de funciones, debe generar patrones de señales periódicas, en función de unos parámetros introducidos por el usuario.





Figura 6.2 Primera aproximación para el generador de funciones

En un principio, un generador de funciones básico puede generar señales senoidales, cuadradas, rectangulares o dientes de sierra, con una amplitud, una frecuencia, un ciclo de trabajo.... Etcétera, pero se ha considerado interesante introducir una funcionalidad nueva al generador de funciones, que es la generación de señales FM.

En telecomunicaciones, la frecuencia modulada (FM) o modulación de frecuencia es una modulación angular que transmite información a través de una onda portadora variando su frecuencia. En aplicaciones analógicas, la frecuencia instantánea de la señal modulada es proporcional al valor instantáneo de la señal moduladora, además de otras muchas aplicaciones.

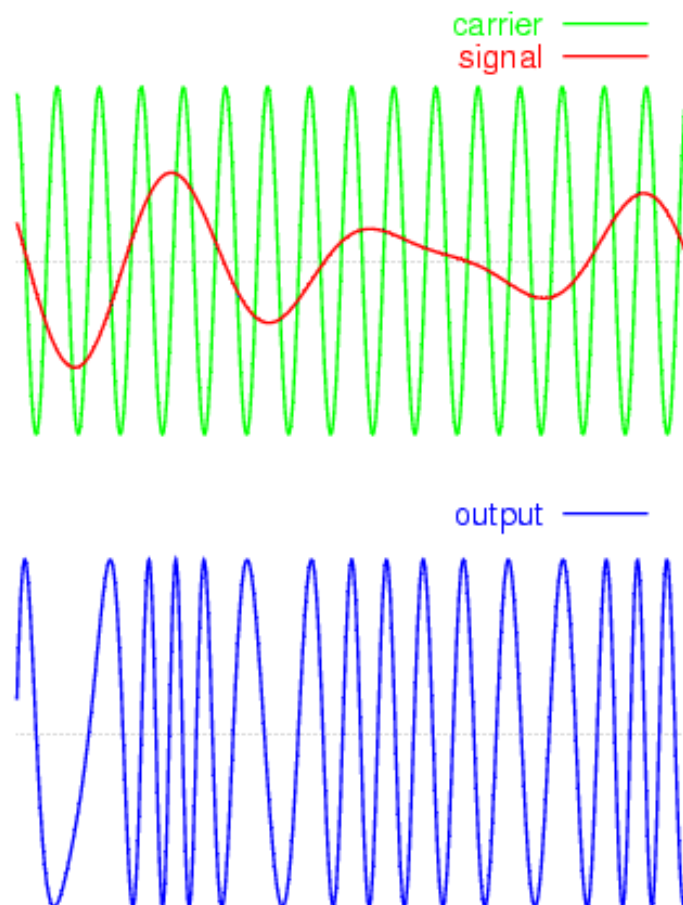


Figura 6.3 Frecuencia modulada

Una vez pensadas las funcionalidades de la aplicación, se presenta un diagrama de bloques más en detalle de lo que será el diseño completo de éste módulo:

### Diagrama de bloques del generador de funciones



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

## Universidad Politécnica de Madrid

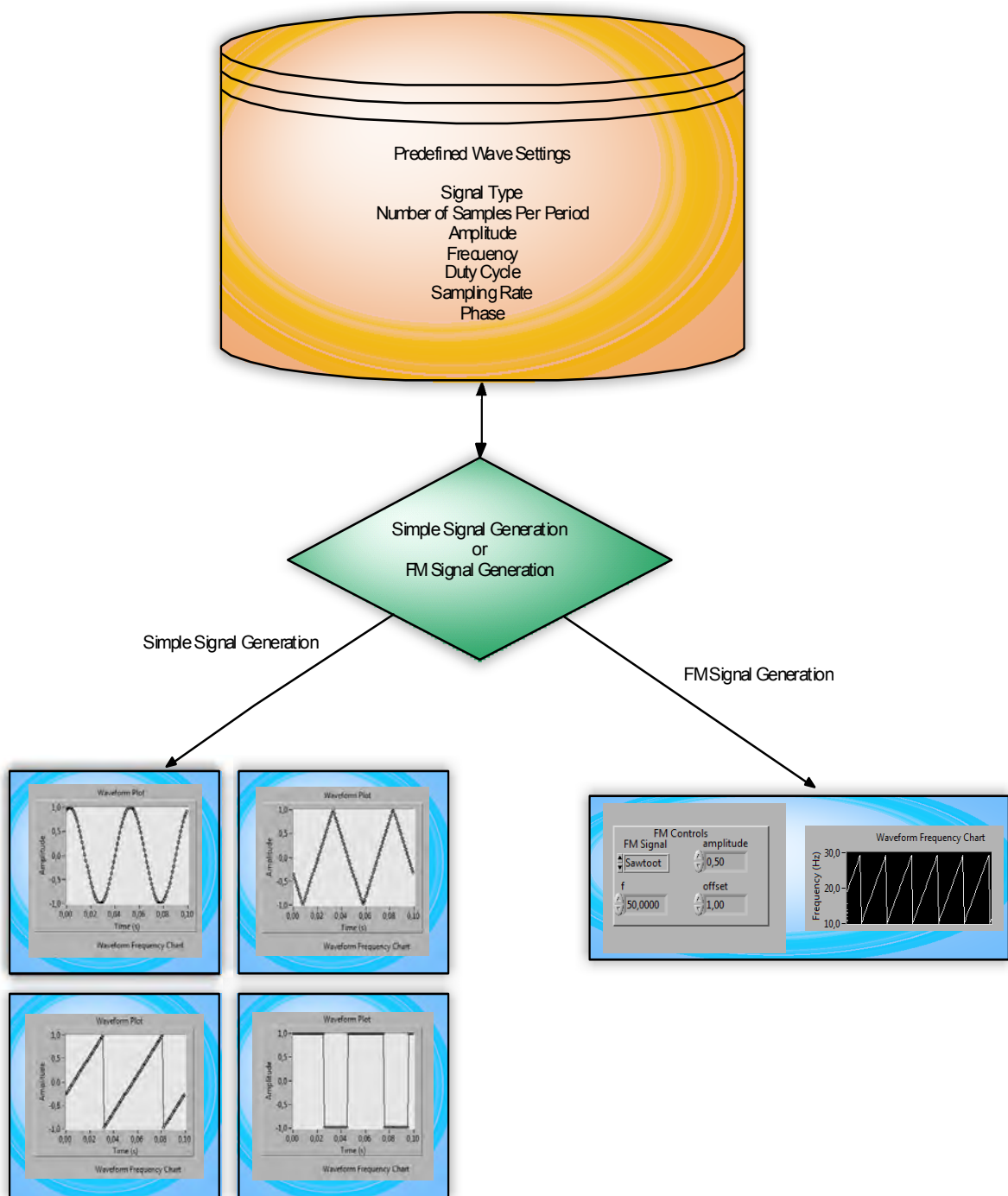


Figura 6.4 Diagrama de bloques generador de funciones





# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

Figura 6.5 Código del generador de funciones

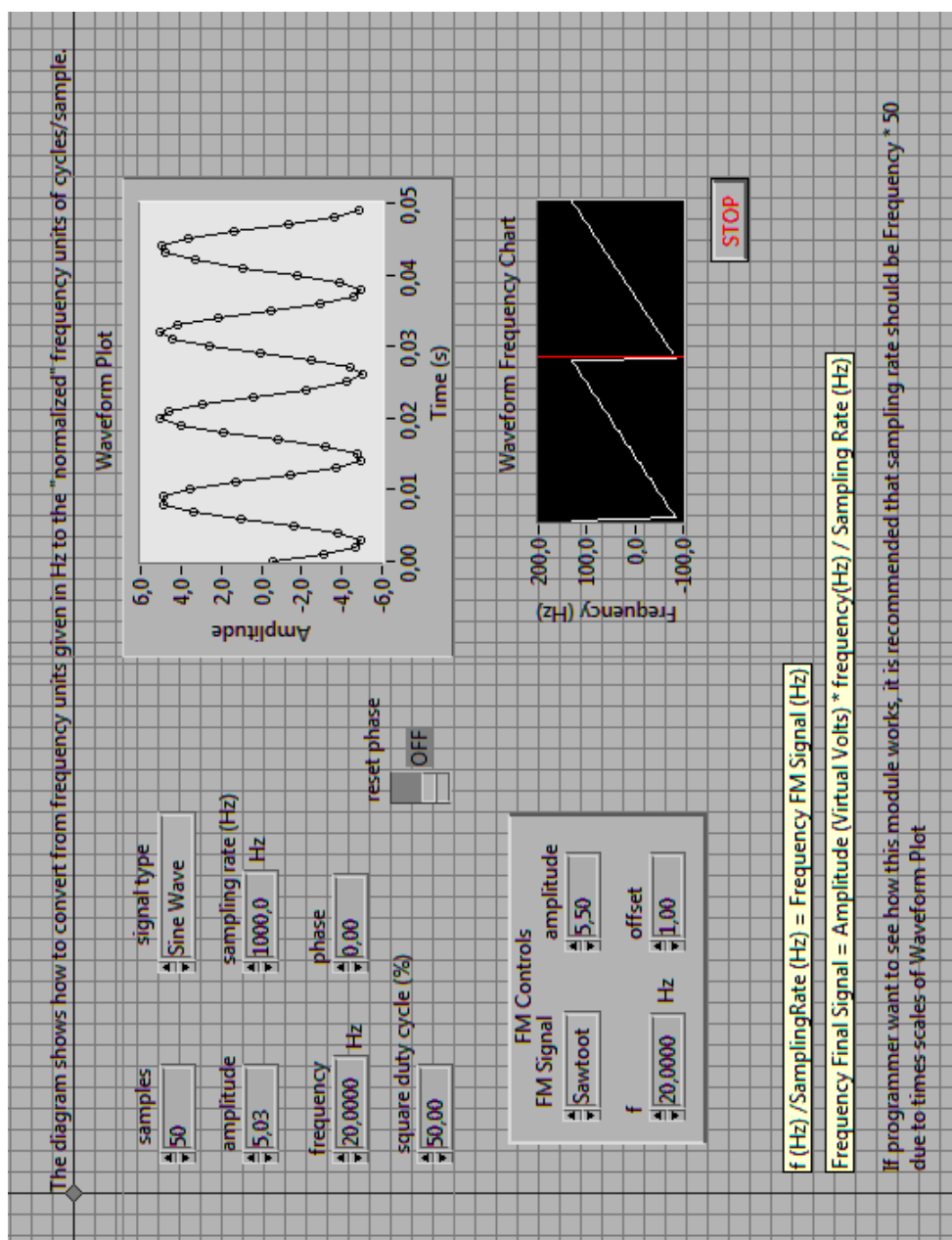




Figura 6.6 Panel frontal del generador de funciones

### Descripción del funcionamiento: generador de funciones

El modulo consta de dos sub-módulos. El primero de ellos genera una señal de tipo senoidal, cuadrada, triangular o diente de sierra, en función de los parámetros de señal configurados por el usuario (salvo la frecuencia, que es función del otro sub-módulo y el usuario). El otro sub-modulo, el de generación de la señal FM, cambia la frecuencia de la señal generada en función del producto de la amplitud de la señal de este sub-modulo por la frecuencia configurada por el usuario. En caso de que el control FM Signal sea igual a cero, este módulo será desactivado, y la frecuencia de la señal generada será únicamente la frecuencia configurada por el usuario.

El usuario debe de tener en cuenta que la frecuencia de la señal generada está normalizada, con el control sampling rate.

Para tener un control detallado de la aplicación, se detalla su funcionamiento y como varían las señales en función de los controles a continuación:

- $X \text{ axys time (secs)} = (\text{samples} / \text{sampling rate}) \text{ (secs)}$
- $\text{Frequency FM Signal (Hz)} = (f / \text{SamplingRate}) \text{ (Hz)}$
- $\text{Frequency Final Signal} = \text{Amplitude (Virtual Volts)} * \text{frequency(Hz)} / \text{Sampling Rate (Hz)}$

Para la codificación de este módulo, se ha utilizado la librería sigxmpl.llb, presente en el LabVIEW 8.5 Professional Development System.

### Analizador de espectro

El módulo software que compone el analizador de espectro debe permitir analizar las frecuencias de la señal de entrada con precisión. Para aumentar la resolución espectral, se introducirán funciones de ventana, además de modos de promediado.



También se considera interesante añadir un módulo de medidas de distorsión.

En principio, el analizador de espectro estará formado por los siguientes bloques:

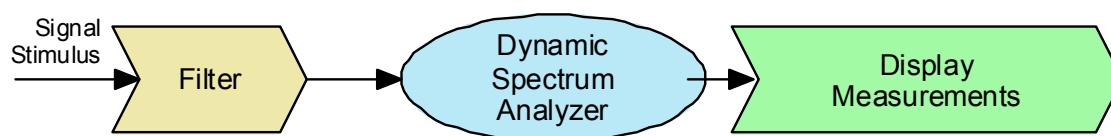


Figura 6.7 Primera aproximación del analizador de espectro

Una vez diseñado el módulo en LabVIEW, se procede a exponer el código del programa:

### **Código y panel del analizador de espectro**



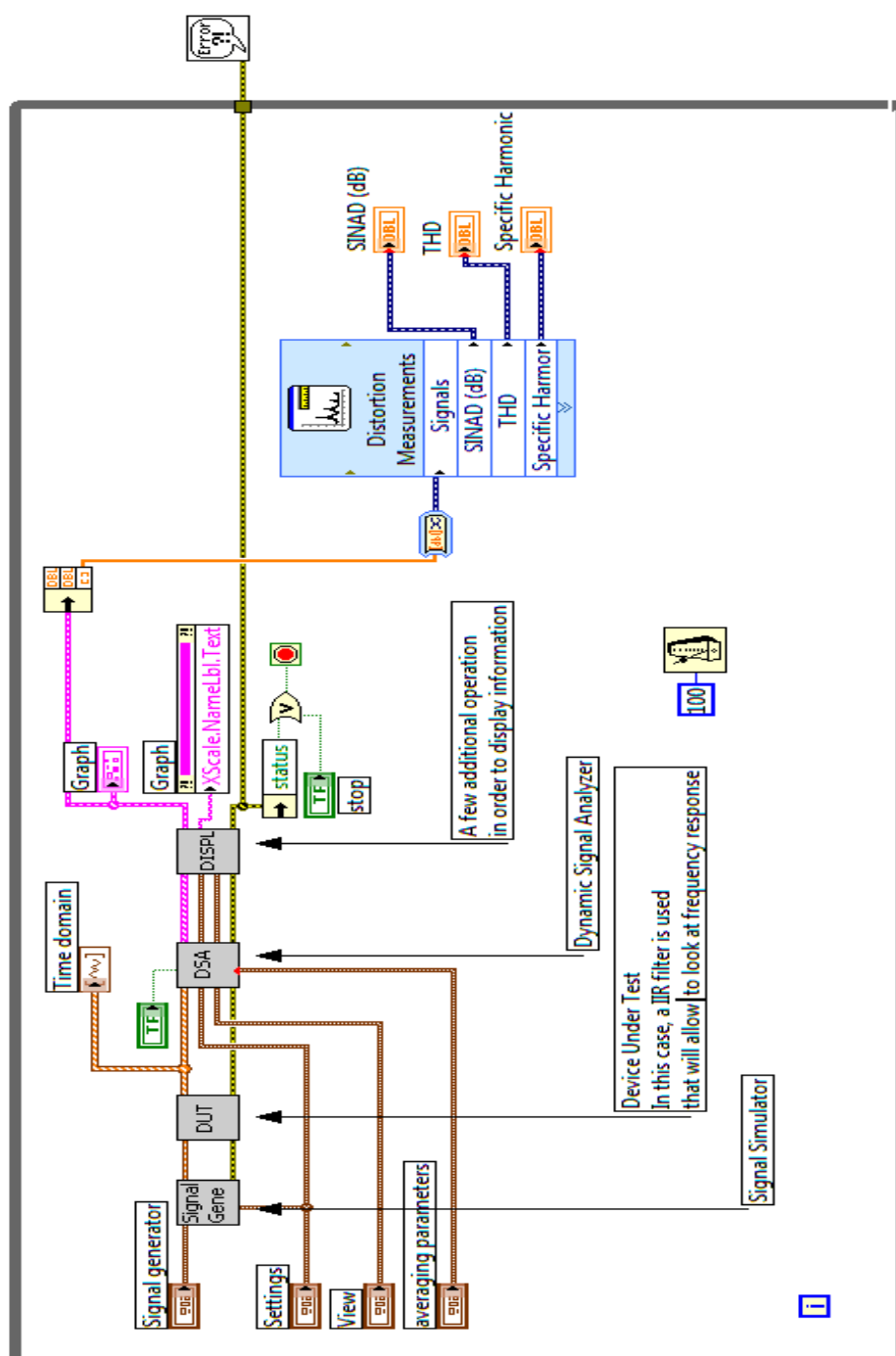


Figura 6.8 Código analizador de espectro



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

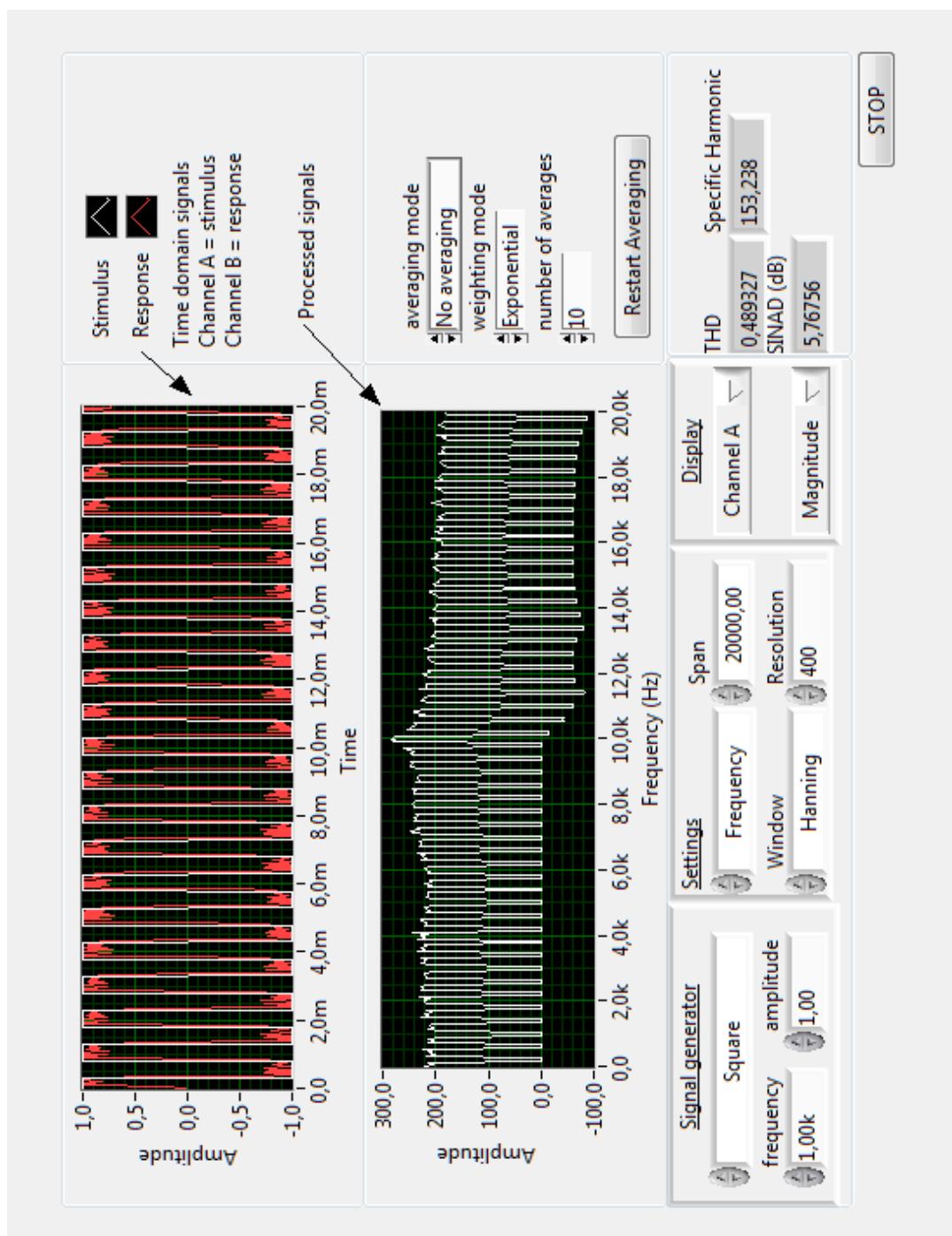




Figura 6.8 Código analizador de espectro

## Descripción de funcionamiento BLOQUE DUT

El bloque DUT (Device Under Test) está formado básicamente por un atenuador de radio-frecuencia (RF attenuator) para proteger la etapa de recepción de señal, en caso de que se reciban señales de amplitudes o frecuencias demasiado altas. Estos atenuadores son usados como adaptador de impedancias y como disipador protector de potencia en medidas de señales de radio-frecuencia. Este atenuador es un filtro IIR paso bajo.

Si la señal se conectase directamente al analizador de espectro sin pasar por el atenuador, posiblemente se vería información adicional de señal

El bloque DUT básicamente construye un array con dos formas de onda a partir de la señal de entrada y cada forma de onda sale por un canal:

- CH0 = Estímulo (la misma señal de entrada)
- Ch1 = respuesta del DUT (salida del filtro IIR)

El filtro IIR permite adoptar las siguientes topologías:

- Off
- Butterworth
- Chebyshev
- Inverse Chebyshev
- Elliptic
- Bessel

También permite configurar la banda de paso del filtro a:

- Low-Pass o paso bajo
- High-Pass o paso alto



- Band-Pass o paso banda
- Band-Stop o banda eliminada

Estos parámetros solo son configurables por el programador, no por el usuario. Solo decir que el filtro IIR está configurado con topología Elliptic paso bajo, ya que:

- Los filtros elípticos exhiben un equi-rizado tanto en banda pasante como en banda eliminada.
- En general, satisfacen los requerimientos del diseño con el menor orden N en comparación con todos los demás tipos de filtro.
- Minimizan la banda de transición.

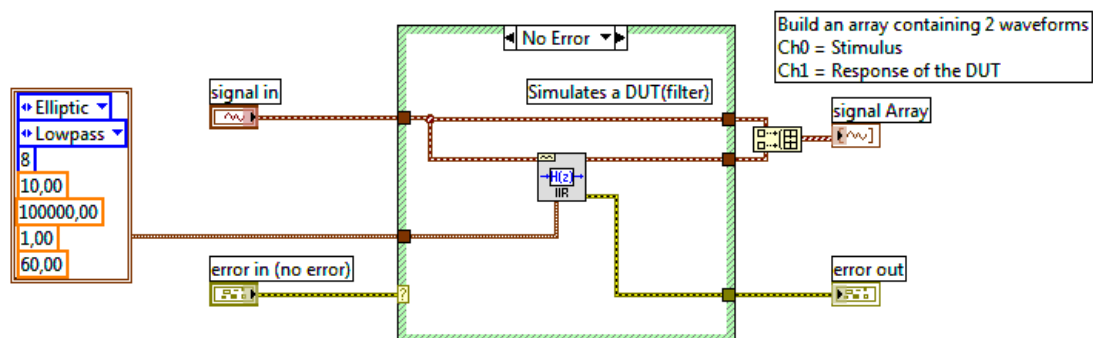


Figura 6.9 Filtro DUT

## Descripción del bloque DSA (Dynamic Signal Analyzer)

Este bloque es el que analiza la señal según le llega. Para el análisis dinámico de la señal, tenemos los siguientes modos:

Scaled Window



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

Se aplica una ventana a la señal de dominio temporal y saca como salidas las constantes de ventana para su posterior análisis.

### FFT Power Spectrum

Calcula el promedio del espectro de potencia de la señal. Este promedio lo calcula según el modo configurado.

### FFT Spectrum (Mag-Fase)

Calcula el promedio del espectro de potencia de la señal, y devuelve la FFT calculada como magnitud y fase.

### Cross Spectrum (Mag-Phase)

Calcula el promedio del espectro cruzado de la señal de entrada. Devuelve los resultados como magnitud y fase. Este promedio lo calcula según el modo configurado.

### Frequency Response Function (Mag-Phase)

Calcula la respuesta en frecuencia y devuelve los resultados como magnitud y fase.

### **Modos de promediado disponibles:**

- Peak – Hold
- RMS
- Exponencial

### **Descripción del bloque Display**

Este bloque únicamente transforma algunas variables para poder mostrarlas correctamente en el panel frontal, por ejemplo, de voltios a decibelios.



## Osciloscopio

El osciloscopio diseñado puede visualizar los canales A, B o A y B. Después de seleccionar los canales, que puede establecer la base de tiempo, V/div, la fuente de disparo, la pendiente y nivel.

En el diagrama de bloques en el código se han dividido las funcionalidades en cuatro bloques, como puede observarse.

El primer bloque empezando por arriba se ocupa de la fuente de disparo (trigger).

El segundo bloque es el de visualización de canales, en función del indicador del panel frontal. La visualización en el Waveform Graph está escalado según los voltios/división y la base de tiempos configurada en los bloques tres y cuatro.

El tercer bloque se ocupa del escalado de la base de tiempos en el eje X del Waveform Graph.

El cuarto bloque se ocupa del escalado de los voltios/división en el eje Y del Waveform Graph.

A continuación se muestra el diagrama de bloques y el panel frontal osciloscopio:

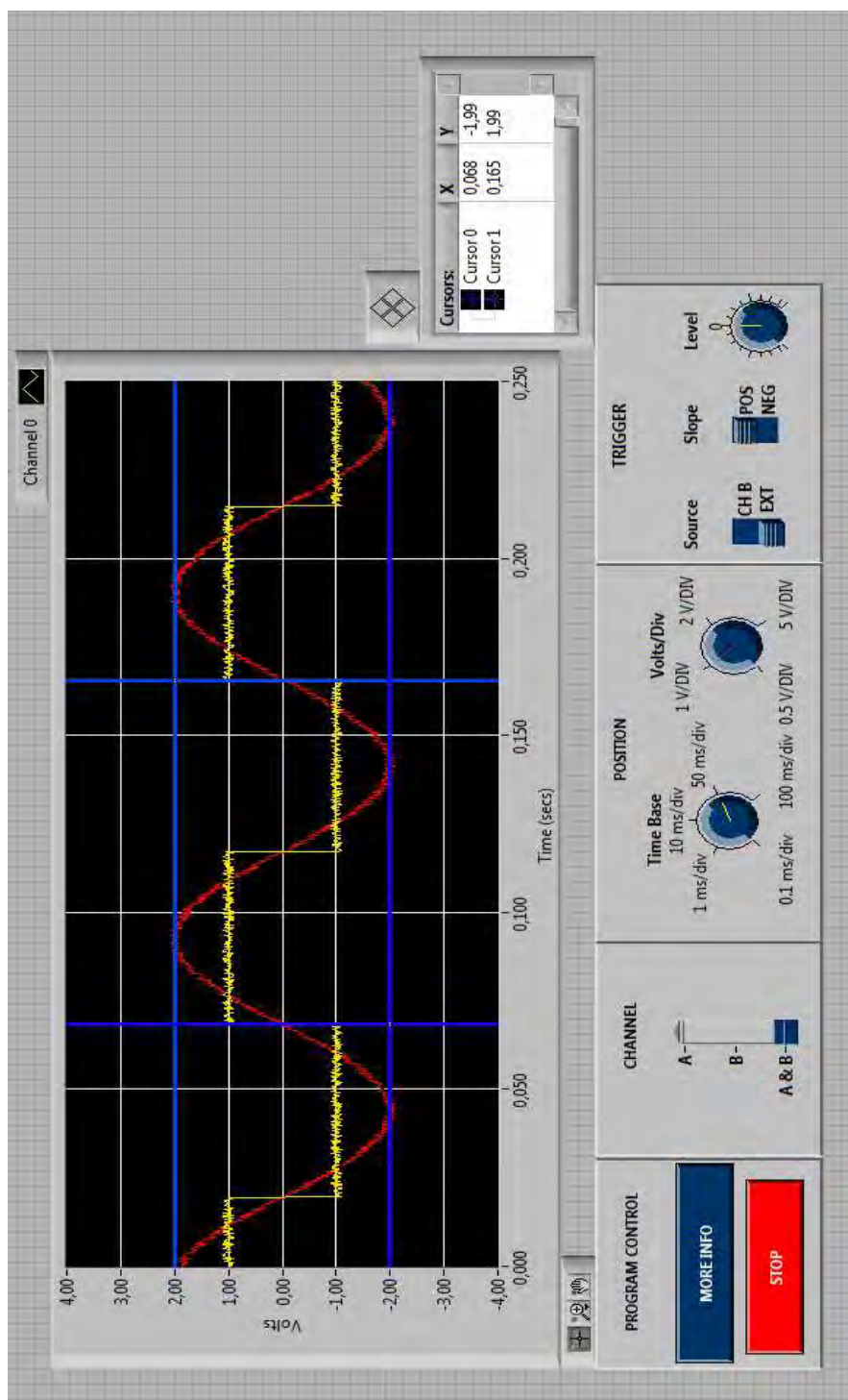


Figura 6.10 Panel frontal del osciloscopio





# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

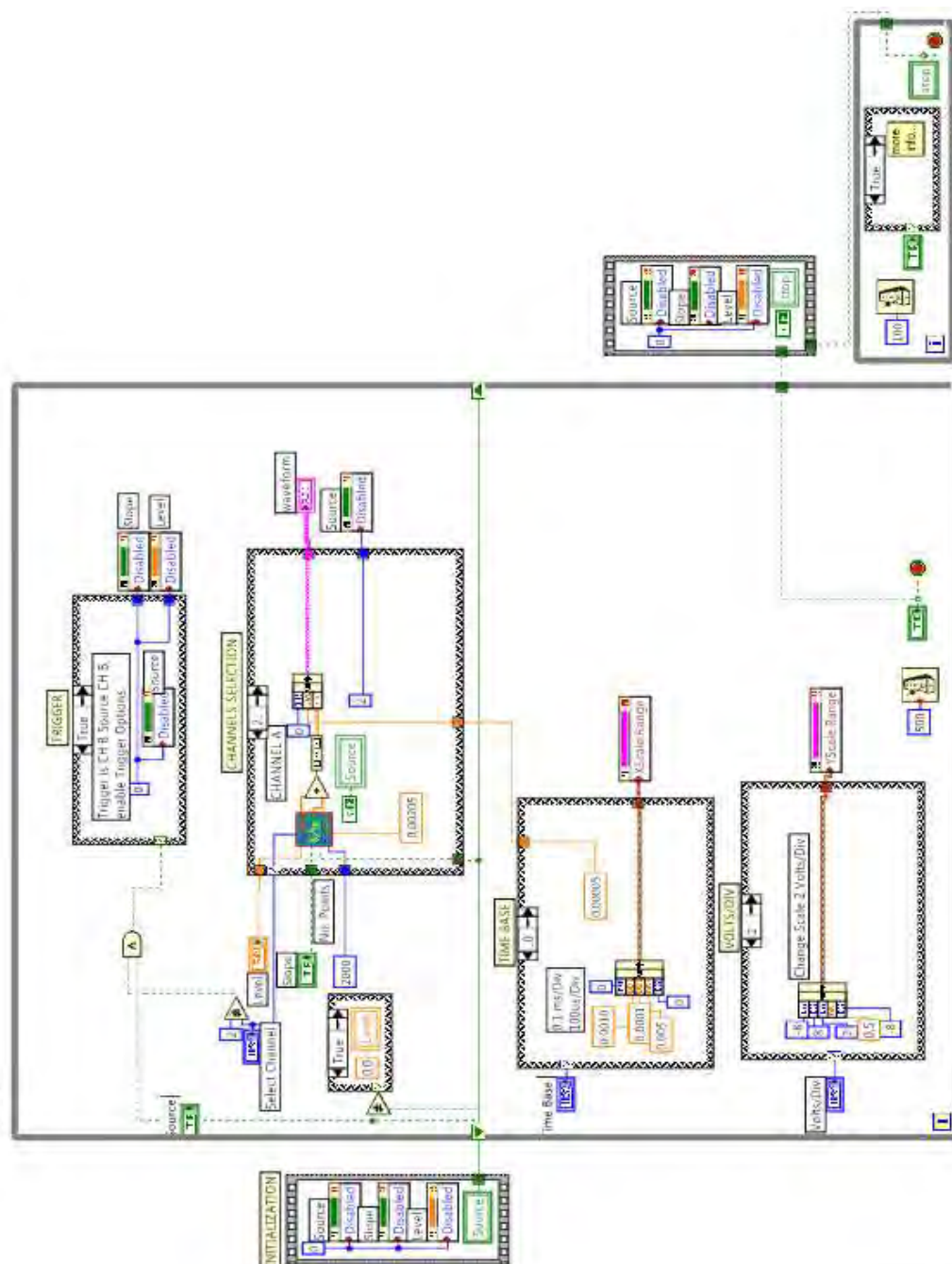


Figura 6.11 Código del osciloscopio



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

Una vez diseñados y probados los tres módulos por separado, se procederá al diseño de la aplicación completa. Como puede observarse, algunas funcionalidades están repetidas en ambos módulos, como la generación de señal, visionado de señales, etc, por lo que esos bloques de código repetido se eliminarán, y se llevará a cabo de tal manera que se utilice el mínimo código posible, para hacer la aplicación mas versátil y que el panel frontal sea sencillo y apetecible para el usuario.



## **Capítulo 7 Integración y test**

### **Introducción:**

Una vez con los tres módulos diseñados por separado, se va a realizar una aplicación generador de funciones, analizador de espectro y osciloscopio, todo en una misma aplicación. Para llevar la aplicación a cabo, se procederá a la reutilización de código, ya que en los bloques diseñados están las funcionalidades que se desean, en la aplicación completa se cogerán 'partes' de código de cada uno de ellos, se juntarán y se probarán funcionando todas conjuntamente. Ya que los bloques por separado están probados, se espera que no requiera mucho tiempo la integración de los sistemas y las pruebas de los mismos.

### **Integración de hardware al sistema: Configuración de la DAQPad-6020E**

El primer paso es probar que el hardware funciona, para ello simplemente basta con conectar un canal, por ejemplo CH0, por el cual se genera una onda senoidal, a un canal de lectura, por ejemplo el CH1. La generación de ondas y la adquisición por los canales de la tarjeta se puede realizar fácilmente configurando la tarjeta a través de los drivers que nos proporciona National Instruments, a través del Software Measurement & Automation Explorer.

Como puede observarse en la siguiente figura, la prueba se realiza con éxito, tanto la generación como la captación de señal.



Figura 7.1 Modo diferencial mejor vs noise



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

## Universidad Politécnica de Madrid

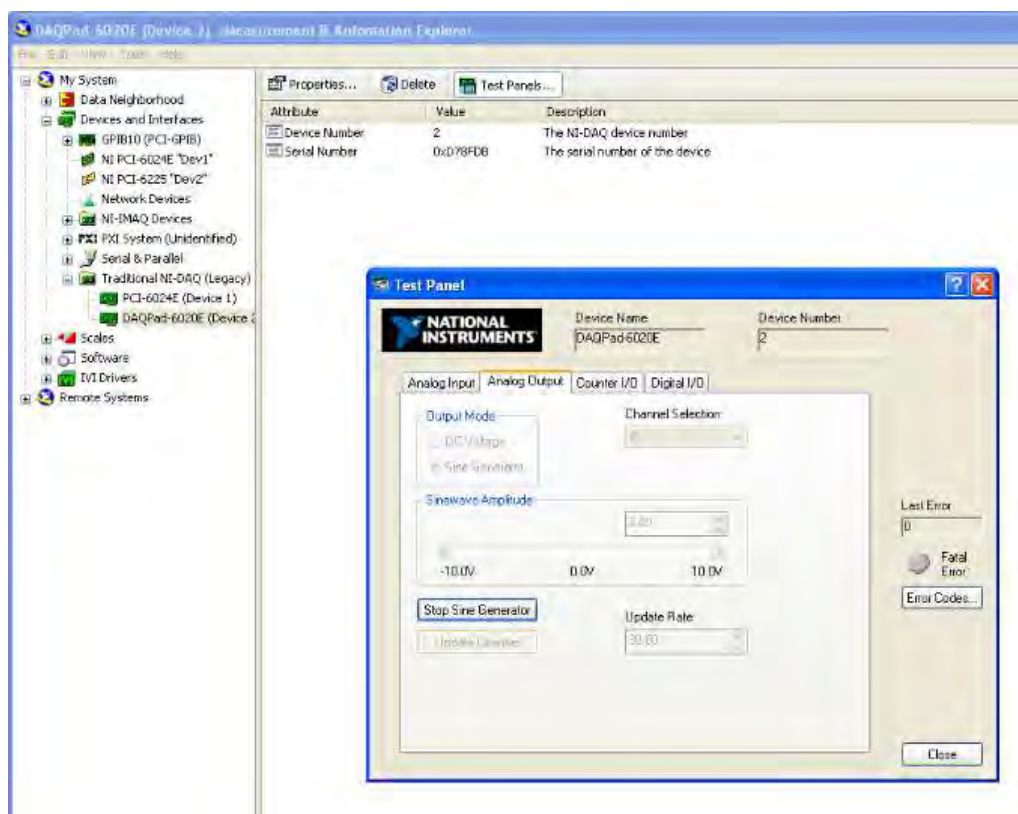


Figura 7.2 Configuración de canal de lectura

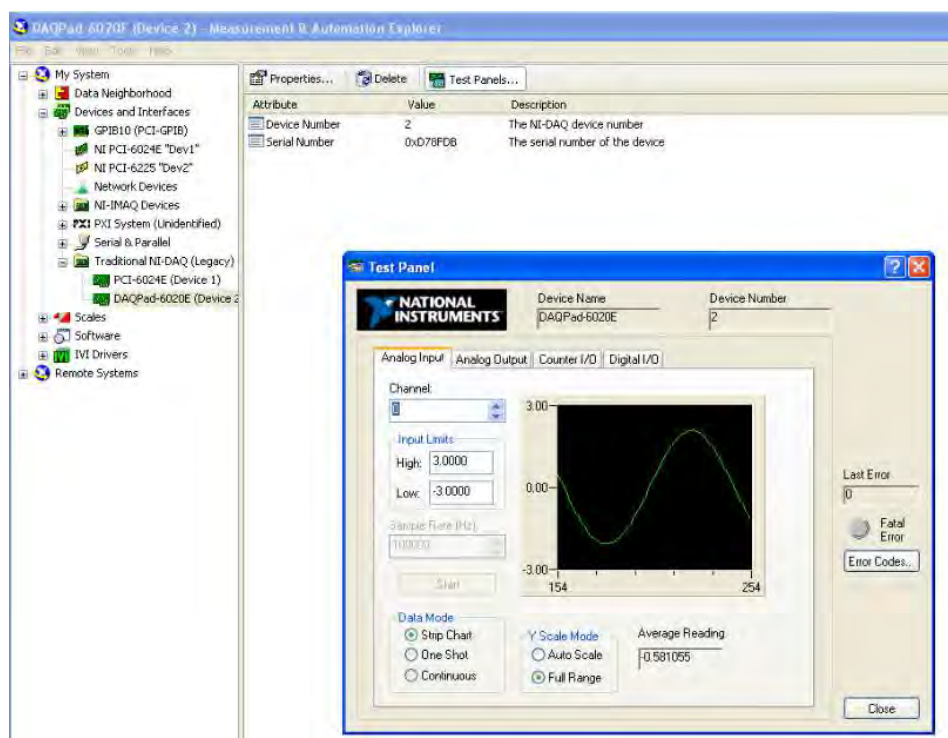


Figura 7.3 Captación de onda senoidal generada conectando CH0 con CH1 lectura-escritura

Por lo tanto, la confirmación del funcionamiento del hardware ya se ha completado.

### Integración del software al sistema: Unión del generador de funciones, analizador de espectro y osciloscopio virtual





# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

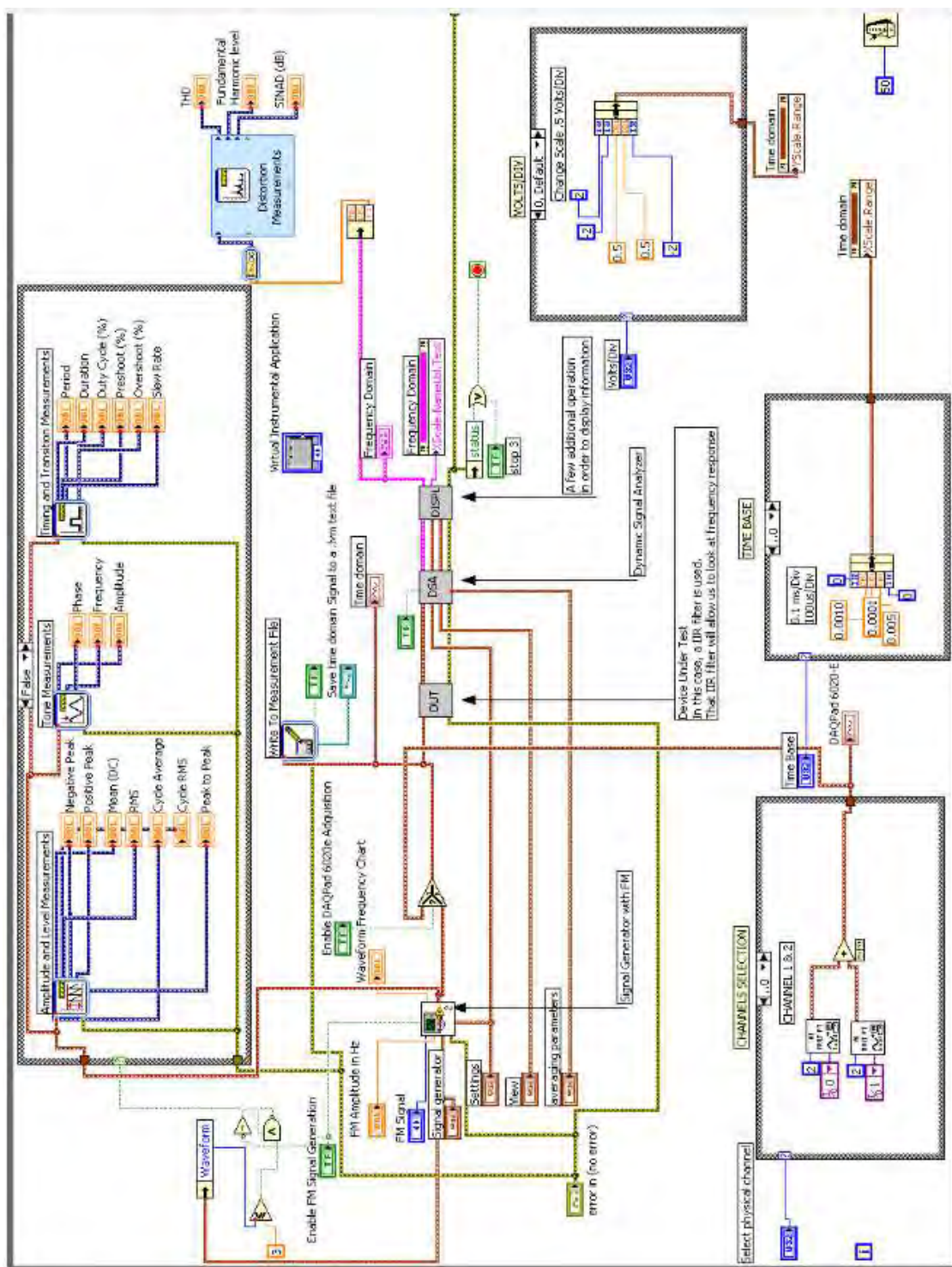


Figura 7.4 Código en LabVIEW de la aplicación completa





Como puede observarse, el bloque base has ido el analizador de espectro. A éste se le han añadido básicamente cinco bloques:

- Generador de funciones con FM (Generador de funciones)
- Eje de tiempos ajustable (osciloscopio)
- Control de escalado de tensión ajustable (osciloscopio)
- Adquisición de datos de dos canales de la tarjeta DAQ
- Medición de parámetros estáticos tipo: amplitud, periodo, componente continua...etcétera.

Todo ello, junto al analizador de espectro, forma la totalidad de la aplicación. Las funcionalidades de ésta se pueden comprobar mejor en el panel frontal, que se muestra en la siguiente sección.

### **Unión del bloque hardware con el bloque software**

La unión del bloque hardware y software se realiza en solo unos minutos. Solo hay que añadir en el código los módulos, o drivers, de lectura y escritura de la tarjeta de adquisición. En este caso, se utilizarán los drivers DAQ-Tradicional. En esta aplicación solo serán necesarios los módulos de lectura-escritura, pero hay otros módulos útiles como módulos de calibración, configuración, etc.

A continuación se muestra el panel Frontal del sistema completo, adquiriendo una señal senoidal externa con la tarjeta de adquisición.



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

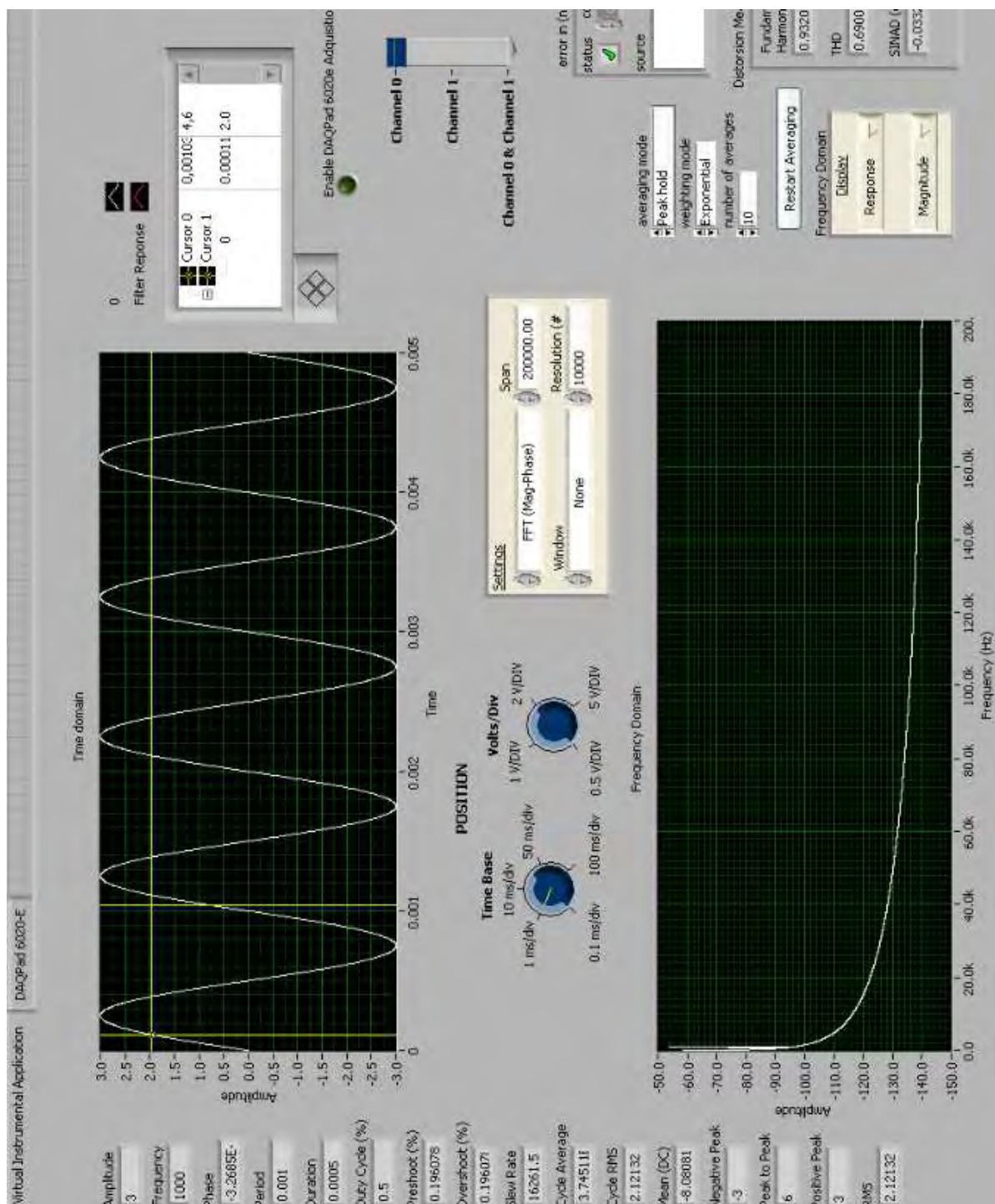


Figura 7.5 Panel frontal, lado izquierdo



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

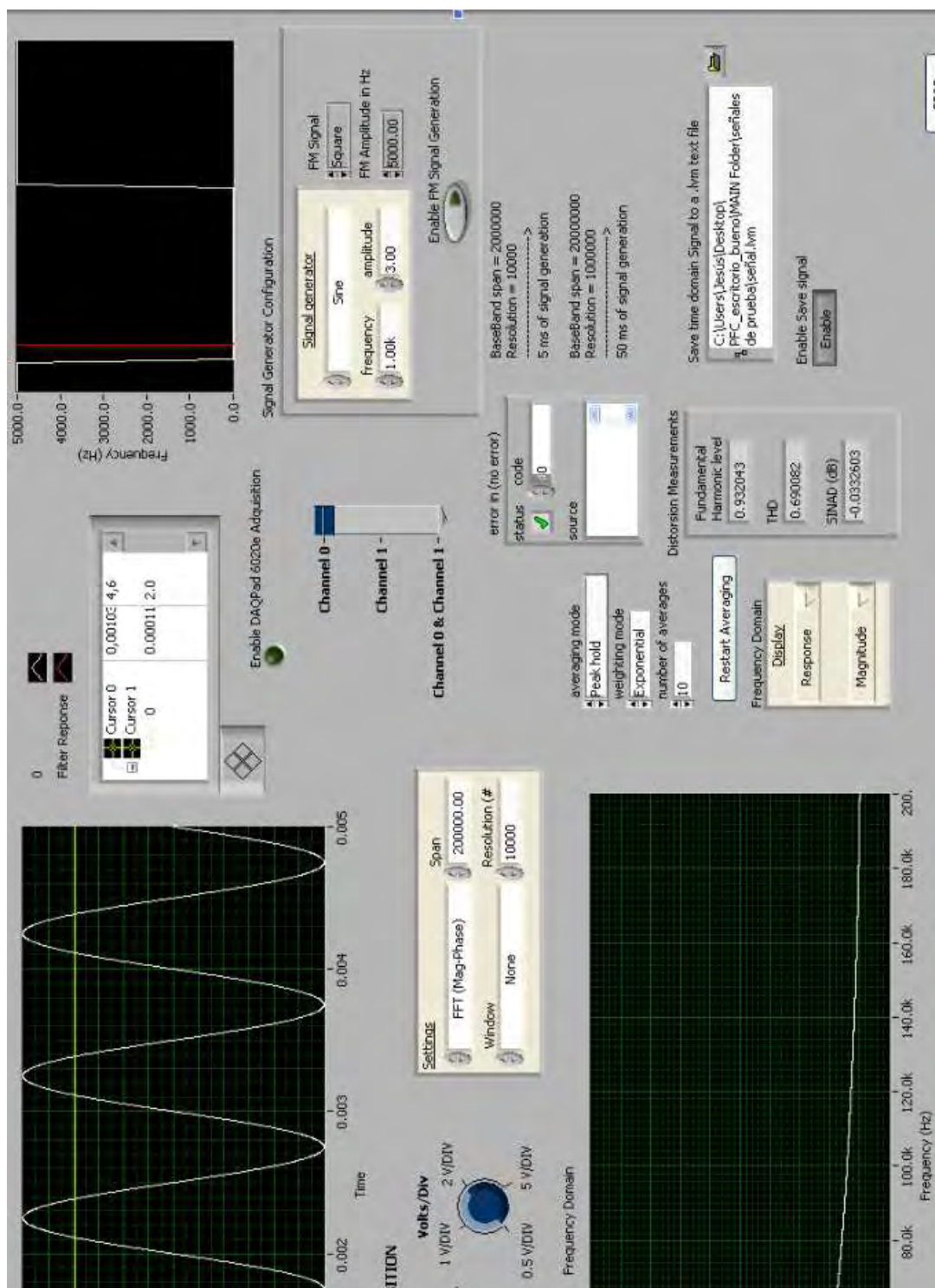


Figura 7.6 Panel frontal, lado derecho





### Zoom de la onda que entra en la aplicación a través la DAQPad-6020E



Figura 7.7 Prueba Seno 1 Khz con tarjeta



## **Capítulo 8: Recursos adquiridos y posibles mejoras**

### **Introducción:**

Los recursos adquiridos con este proyecto básicamente se componen de dos vertientes: el código, el cual se podrá reutilizar para futuros proyectos, y la experiencia adquirida por ingenieros que han trabajado en él.

El proceso de industrialización del software, exige que los ingenieros y técnicos planteen nuevas alternativas para incrementar la productividad de los desarrolladores y analistas en el desarrollo de sistemas de software.

En este contexto, hay estudios que evalúan la productividad de un desarrollador realizando una comparación entre un sistema desarrollado reutilizando software, y otro sin reutilización de software. Para ello, en la evaluación se utiliza el Método Aproximativo de Costo y Productividad (COCOMO), en el cual no se entrará en detalle en éste proyecto.

### **Reutilización de Software**

Mientras que los usuarios demandan mayor complejidad y funcionalidad de los sistemas de software, los desarrolladores por su parte, tienen que manejar la complejidad propia de la lógica de negocios, así como la complejidad de la herramienta de programación, con los limitados recursos económicos y de tiempo. En este escenario, la reutilización de software, se constituye como una de las mejores opciones para disminuir los plazos de entrega.

Existen diferentes técnicas de reutilización sistemática de software (4), dentro del esquema de la programación orientada a objetos (POO) uno de ellos es la reutilización de componentes, a través de la creación de clases abstractas, e interfaces.

### **Recursos obtenidos con la realización del proyecto:**



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

La aplicación software ha constado de:

- Un generador de funciones virtual

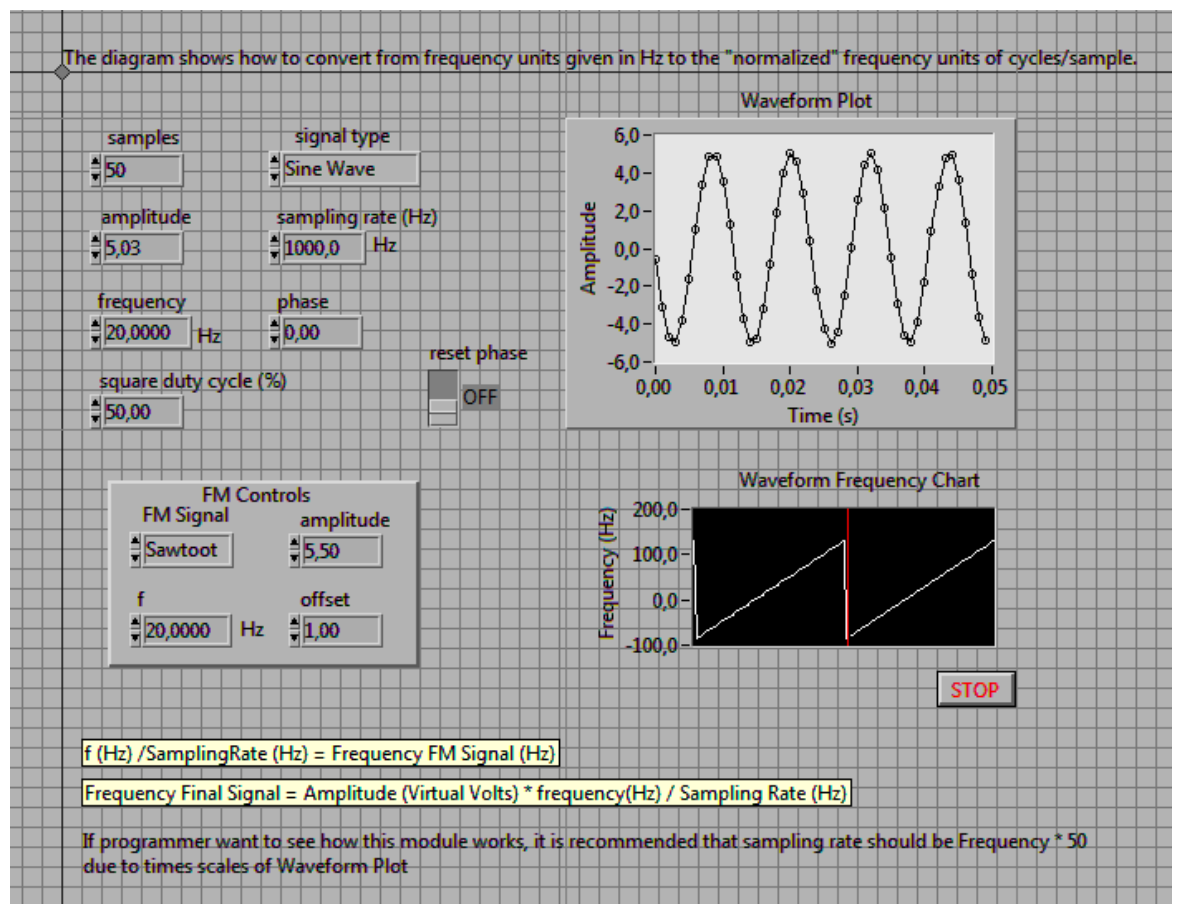


Figura 8.1 generador de funciones virtual

- Un analizador de espectro virtual



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

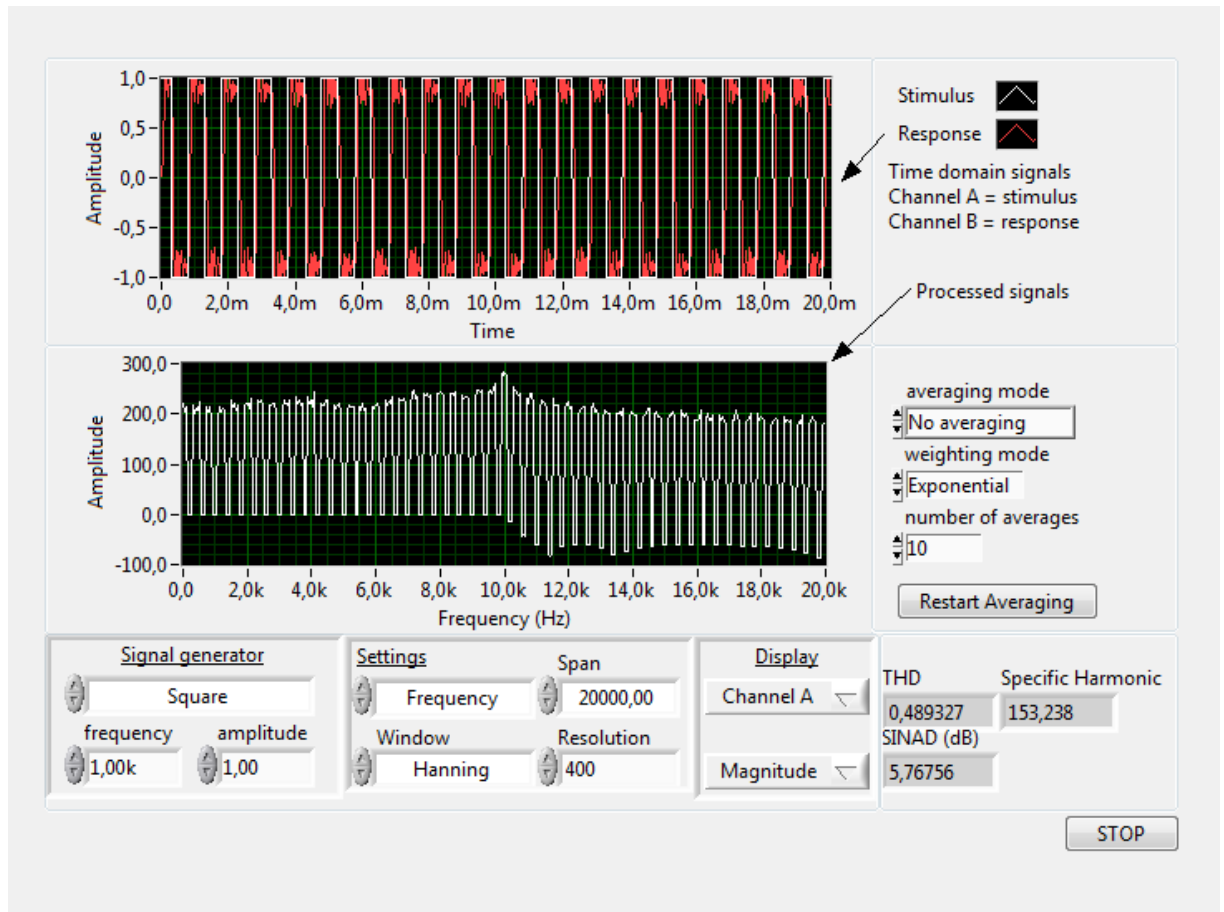


Figura 8.2 Un analizador de espectro virtual

- Un osciloscopio virtual





# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

## Universidad Politécnica de Madrid

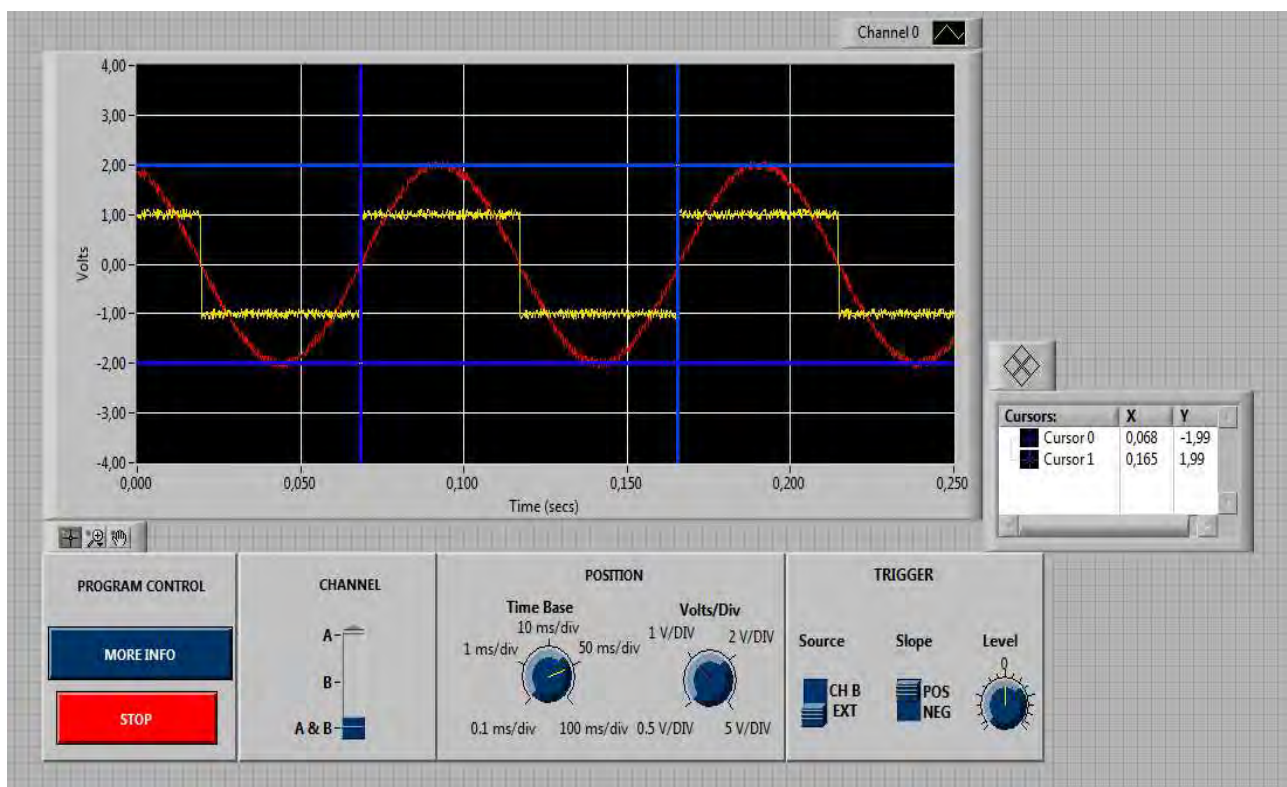


Figura 8.3 Un Osciloscopio virtual

- Aplicación que incluye los tres bloques juntos



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

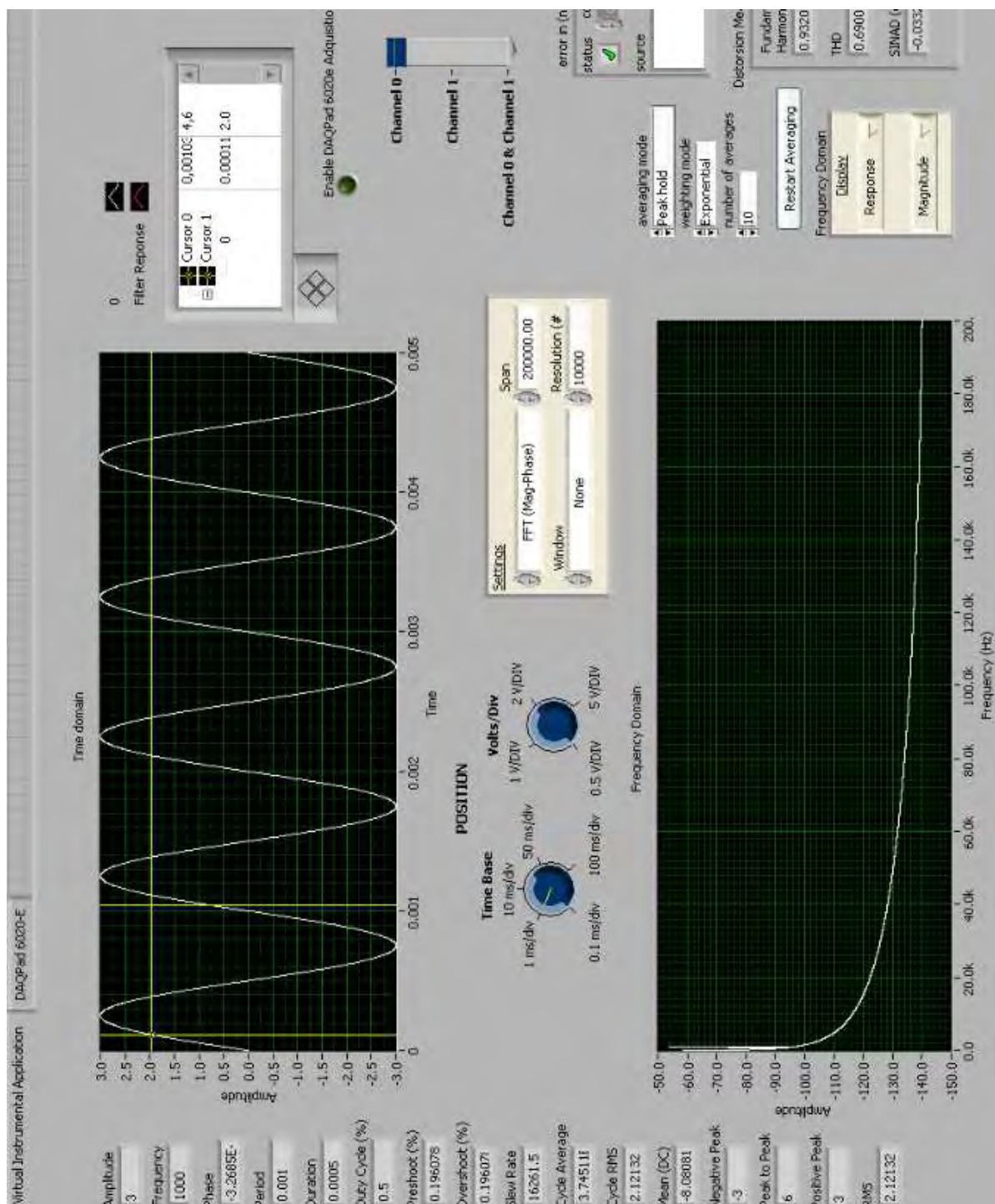


Figura 7.5 Panel frontal, lado izquierdo



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

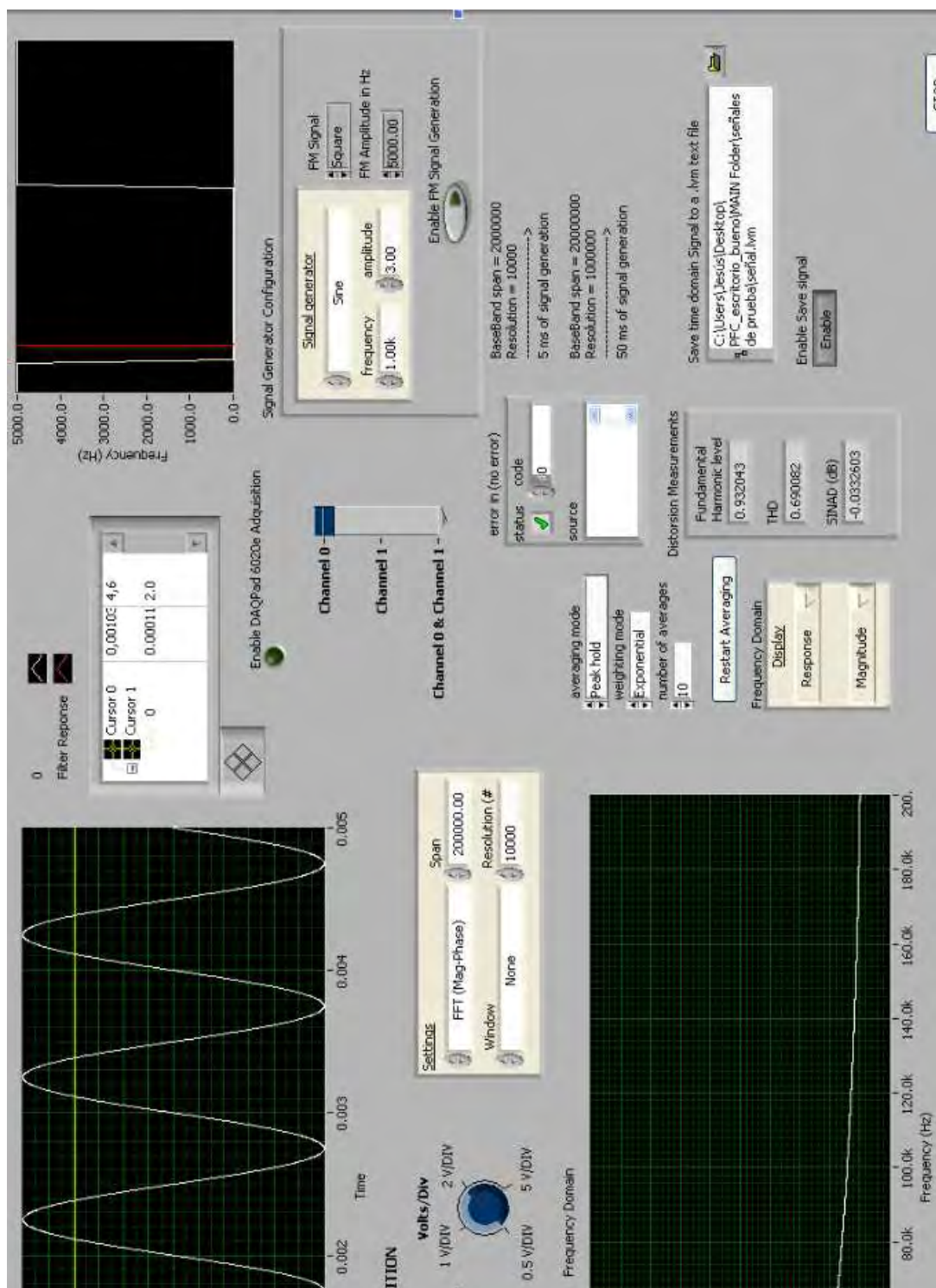


Figura 7.6 Panel frontal, lado derecho





Estos bloques se podrán utilizar como librerías o como partes de futuros módulos más grandes en futuros proyectos.

Para evaluar la productividad en el desarrollo de un sistema reutilizando el código, es necesario desarrollar el mismo sistema con la misma funcionalidad pero sin reutilización del código.

### **Conclusiones: método COCOMO**

Según el cálculo utilizando el método aproximativo de costo y productividad (COCOMO) realizado en multitud de estudios de economía de empresa, en futuros proyectos se observaría una reducción de aproximadamente un 43,89% en el plazo de entrega si se reutiliza software a través de componentes, así como un ahorro del 78,15% en los costos de mano de obra.

La reutilización de software, a través de componentes, se constituye como una forma de mejorar la productividad en el desarrollo de sistemas de software.



### Posibles mejoras:

**Ancho de banda:** Los test en la aplicación muestran que para señales de 100 KHz, el sistema deja de ser preciso gradualmente. En el futuro se deberá estudiar cómo mejorar ese hecho.

Posible idea: Trabajar con frecuencias altas requeriría una tarjeta de adquisición más moderna, la actual ya está bastante anticuada según las disponibles en el mercado actual. Aparte, el filtro IIR de la aplicación está configurado como paso bajo, cuya frecuencia de corte es de 100KHz.

**Lentitud de la aplicación:** Esta aplicación utiliza numerosas funciones matemáticas que proporciona LabVIEW. Al hacerlo todas a la vez, empieza a ralentizarse cuando se generan señales de frecuencia mayor a 50 KHz. Se deberá estudiar el cómo separar las tareas, para no saturar al procesador.

Posible idea: dividir los bloques y tratarlos como si fuesen hilos en C++, trabajando en paralelo. Esos hilos irán dentro de un bucle while, cada uno, y se ejecutarán continuamente salvo que se produzca un error, pero con un timestamp de 100 ms para que cada hilo realice los cálculos cada 100 ms, y no continuamente. Este timestamp es una estimación a priori, se deberá probar con diferentes tiempos y comprobar los resultados, porque también existe una variable adicional, que es la velocidad del procesador/es.

**Más formatos para guardar las señales adquiridas:** puede ser interesante presentar las señales guardadas en diferentes formatos, para que sean compatibles con diversas aplicaciones, como programas de análisis matemáticos más especializados en la materia, o por otros osciloscopios y/o analizadores de espectro comerciales, o para programas de otros sistemas operativos, como pueden ser los sistemas operativos UNIX. Esta mejora daría posibilidad a expandir el producto a más fuentes de mercado, ya que sería utilizable por más usuarios.

**Utilización de más canales de la tarjeta:** la tarjeta DAQPad 6020-E posee más de ocho canales, de los cuales la aplicación solo usa tres. Dos para adquisición y uno para



generación de señal. Sería interesante que la aplicación contase con la posibilidad de añadir más canales.

**Aislamiento galvánico / Optoacopladores:** añadir aislamiento galvánico u optoacopladores a los sistemas de medida es una buena opción si queremos que el sistema de medida sea lo más preciso posible. Se deberá dar la opción al cliente de poder añadir esta funcionalidad en caso de que la utilización de la aplicación sea para proyectos militares o muy profesionales.

## Introducción de hardware entre señal y sistema de medida

### **Pequeña introducción al aislamiento y su problemática:**

En una aplicación de medida, la tensión en modo común no es objeto de medida, por lo que nos interesa un CMRR alto. Pero además hay que tener en cuenta la máxima tensión en modo común que puede soportar el amplificador sin dañarse. Este límite no suele andar lejos del valor de la tensión a la que se alimenta el amplificador. Si la masa de la señal y la del sistema de medida son distintas (p.Ej. están alejadas), existirá una diferencia de potencial que provocará una corriente entre ambos y por tanto un error en la medida.

### **Optoacopladores**

Un optoacoplador combina dos dispositivos semiconductores: fotoemisor y fotoreceptor existiendo entre ambos un camino por donde se transmite la luz. Estos elementos se integran dentro de un encapsulado que por lo general es del tipo DIP para dar cabida a dichos elementos.



Los optoacopladores son capaces de modular una señal luminosa partiendo de una señal eléctrica para luego convertirla otra vez en señal eléctrica. De esta forma, establecen un aislamiento galvánico entre los circuitos de entrada y salida.

- La señal de entrada es aplicada al fotoemisor (LED) y la salida se toma del fotoreceptor (fotodiodo, fototransistor).
- Los fotoemisores que se emplean en los optoacopladores de potencia son diodos que emiten rayos infrarrojos (llamados IRED: Infrared Emitting Diode) y los fotoreceptores pueden ser tiristores o transistores.

Cuando aplicamos una tensión sobre los terminales del diodo IRED, éste emite un haz de rayos infrarrojos que se transmite a través de una pequeña guía-ondas de plástico o cristal hacia el fotoreceptor. La energía luminosa que incide sobre el fotoreceptor hace que este genere una tensión eléctrica a su salida.

Aislamiento Galvánico: es la ausencia de cualquier camino directo para corriente DC.





**Escuela Universitaria de Ingeniería  
Técnica de Telecomunicación**

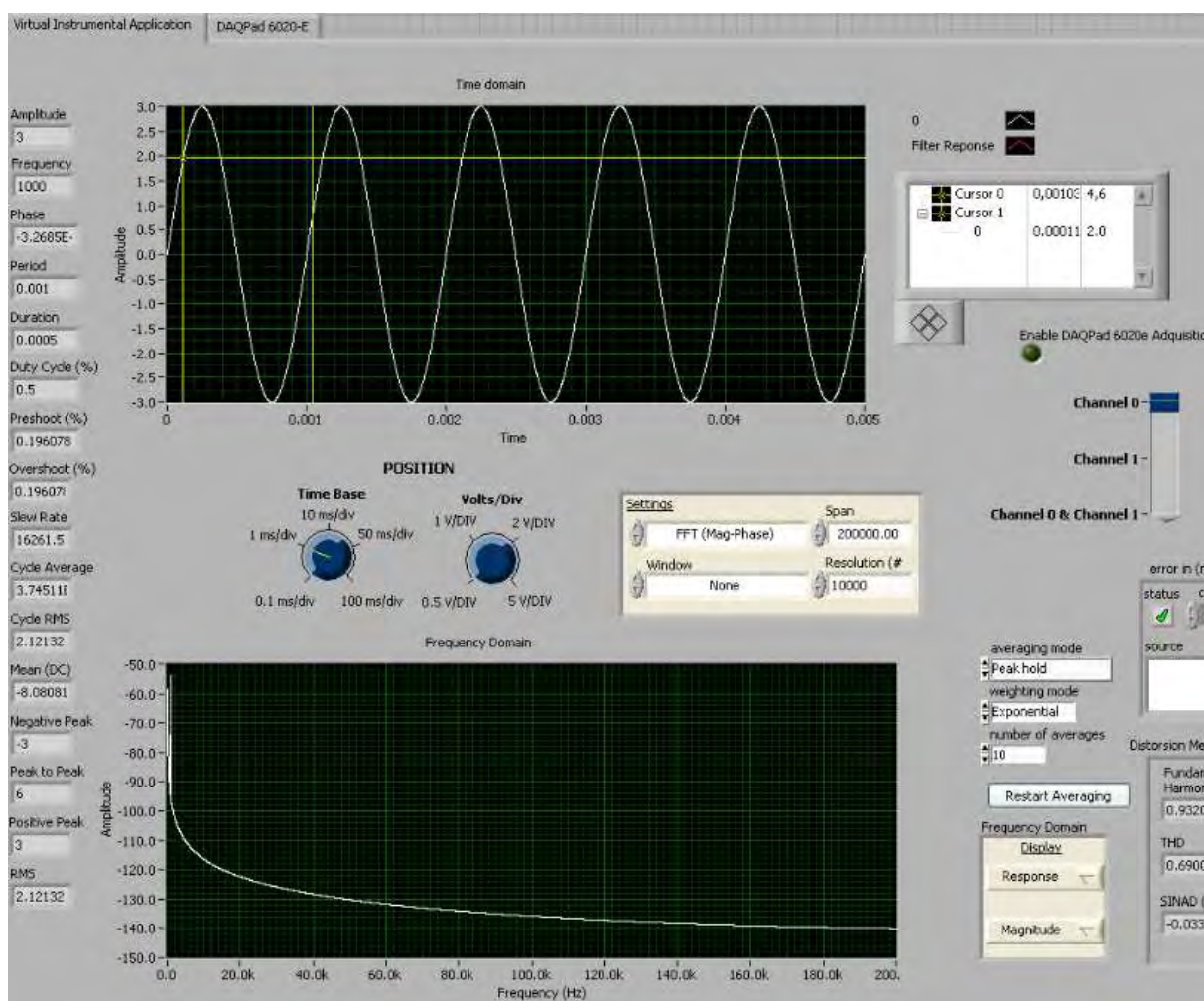
**Universidad Politécnica de Madrid**



## Capítulo 9 Manual de usuario

### Introducción:

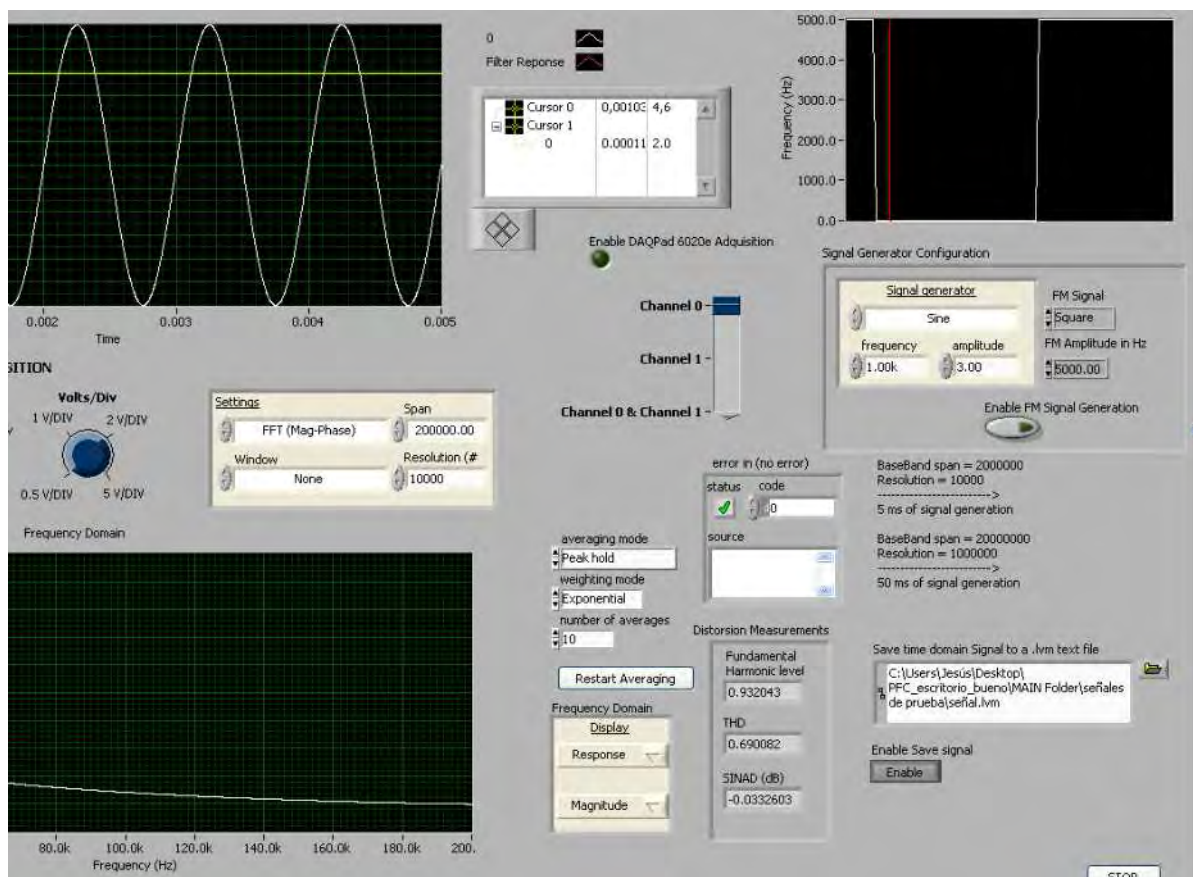
Es interesante proporcionar al usuario una explicación ilustrativa de los indicadores del panel frontal del sistema completo, para introducirle al manejo de la aplicación lo antes posible. Con este pequeño manual explicativo y el capítulo de fundamentos teóricos, el usuario debería de ser capaz de entender en su totalidad, o casi en su totalidad el sistema completo





# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

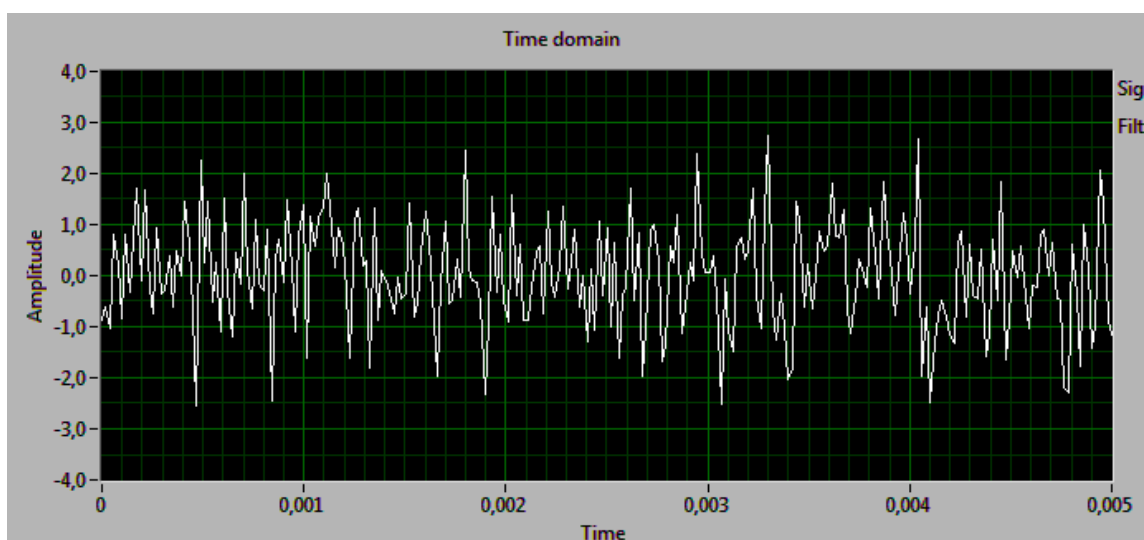
Universidad Politécnica de Madrid



## Componentes y controles del sistema:

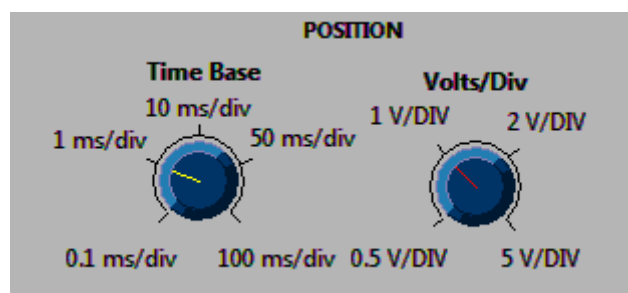


### Time Domain WaveForm Graft:



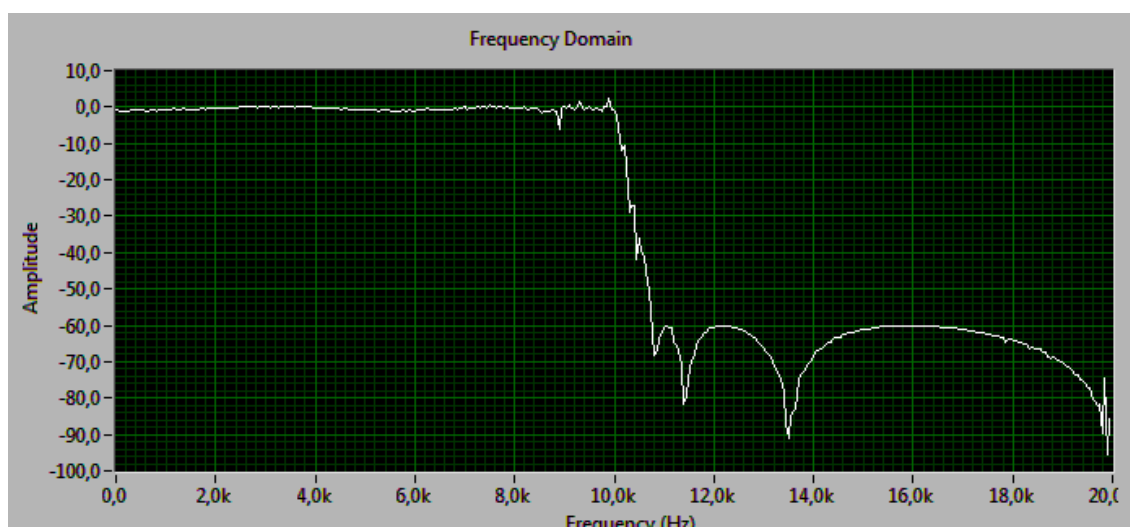
Gracias a esta gráfica podemos observar el dominio temporal de la señal, ya sea la adquirida por la DAQPad 6020-E o la señal generada por el generador de funciones.

### Escalado de tiempos y voltios por división de la gráfica de tiempos:





### Frequency Domain WaveForm Graft:

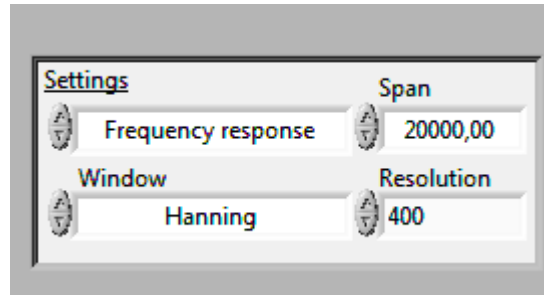


Esta gráfica muestra todas las funcionalidades del analizador de espectro, ya sea inventariado, respuesta en frecuencia, transformada de Fourier, etcétera.



# Escuela Universitaria de Ingeniería Técnica de Telecomunicación

## Universidad Politécnica de Madrid



**Settings:**

- Windowed time
- Power Spectrum
- ✓ FFT (Mag-Phase)
- Cross Power spectrum
- Frequency response

**Window:**

- None
- ✓ Hanning
- Hamming
- Blackman-Harris
- Exact Blackman
- Blackman
- Flat Top
- 4 Term B-Harris
- 7 Term B-Harris
- Low Sidelobe





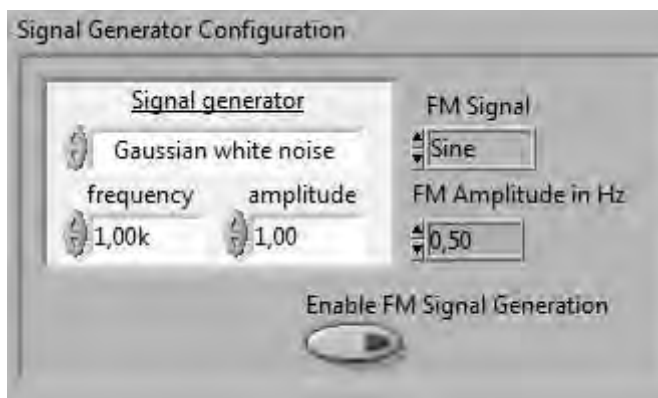
Se ofrecen ocho tipos de enventanado, o la posibilidad de no utilizar esta función.

### Span & Resolution:

Duración y resolución de señal generada: hace referencia al número de muestras de la señal, y la frecuencia de muestreo en la generación de la misma. No hace falta que el usuario tenga conocimiento exacto de estos parámetros, ya que se ha realizado una conversión en el código para que la duración y resolución de la señal generada siga los siguientes patrones:

```
BaseBand span = 2000000  
Resolution = 10000  
----->  
5 ms of signal generation  
  
BaseBand span = 20000000  
Resolution = 1000000  
----->  
50 ms of signal generation
```

Se recomienda al usuario seguir este patrón de generación de señal, con una duración 200 veces mayor que la resolución.





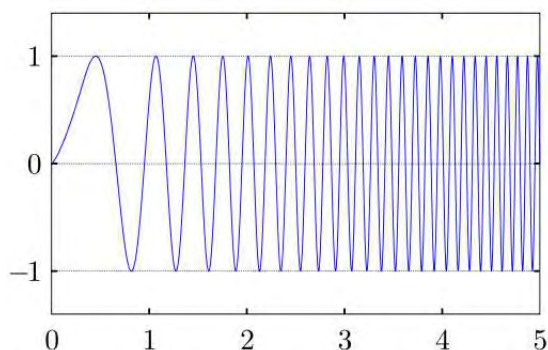


### Signal Generator:

- Sine
- Square
- Triangle
- Sawtooth
- Chirp
- ✓ Uniform White noise
- Gaussian white noise
- PRN

La aplicación ofrece la posibilidad de generación de formas de onda senoidal, cuadrada, triangular, diente de sierra, Chirp, ruido blanco uniforme, ruido blanco Gaussiano y PRN. A continuación se hará un breve recordatorio de la forma de onda Chirp y PRN, ya que puede que sean desconocidas para el usuario.

### Chirp:



Se utiliza comúnmente en el sonar y radar, pero tiene otras aplicaciones, como por ejemplo en las comunicaciones de espectro disperso.



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

En óptica, pulsos láser ultracortos también exhiben formas Chirp, que, en sistemas de transmisión óptica, interactúa con las propiedades de dispersión de los materiales, aumentando o disminuyendo la dispersión total del pulso.

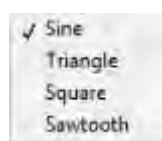
PRN:

En criptografía, el ruido pseudo-aleatorio (PRN) es una señal similar al ruido que satisfaga uno o más de las pruebas estándar de aleatoriedad estadística.

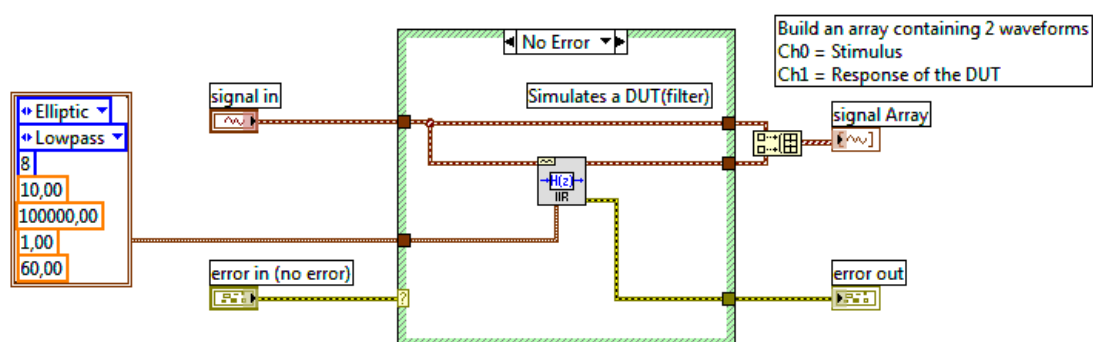
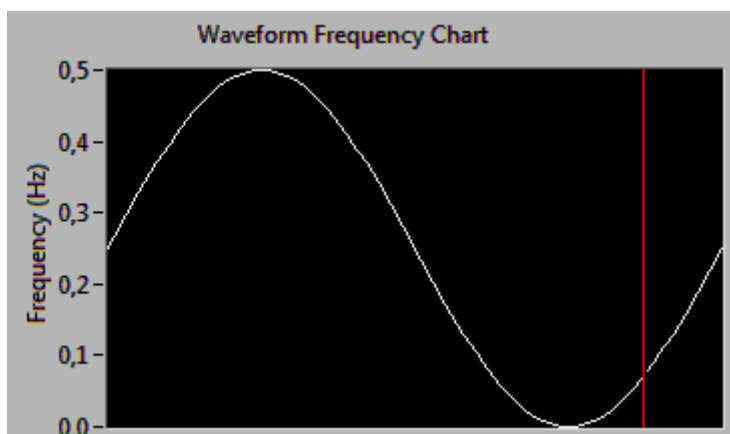
Aunque parece que carecen de cualquier patrón definido, el ruido pseudo-aleatorio consta de una secuencia determinista de pulsos que se repetirá después de su período.

Se ha considerado interesante integrar esta señal al generador de funciones porque los receptores GPS correlacionan el flujo de PN que reciben, con una referencia local para medir la distancia.

### FM Signal:



La aplicación puede generar señales cuya frecuencia es variable en función de la amplitud actual de otra señal (configurable por el usuario). Cómo va variando la frecuencia de la señal generada se muestra en un Waveform Chart. El usuario tiene la opción de seleccionar la amplitud pico a pico (en Hz) de la señal, aunque el límite inferior siempre será 0 Hz. El usuario también será capaz de configurar éste tipo de señal, siendo posibles una senoidal, triangular, cuadrada o diente de sierra.

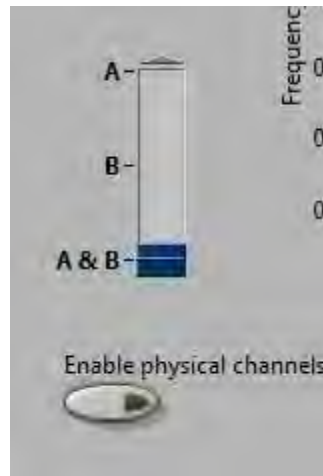


Recordar que el filtro IIR del bloque DUT tiene una frecuencia de corte de 100 KHz, por lo que se recomienda no trabajar con frecuencias mayores a 100KHz.

### Channels Selection:



## Escuela Universitaria de Ingeniería Técnica de Telecomunicación Universidad Politécnica de Madrid

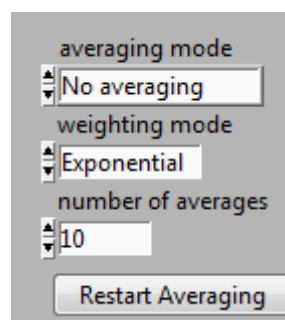


Se ofrece la posibilidad de seleccionar el canal A (Canal CH0) o, el canal B(Canal CH1) o la suma de ambas señales de ambos canales, para mostrarlo en la gráfica de dominio del tiempo.

### **Enable Physical channels:**

La adquisición de señales a través del canal CH0 y CH1 también será habilitada por el usuario. Si este interruptor es habilitado, inmediatamente la aplicación recoge la/las señales de la tarjeta, y no las del generador de funciones.

### **Averaging Mode:**





## Escuela Universitaria de Ingeniería Técnica de Telecomunicación

Universidad Politécnica de Madrid

- ☒ No averaging
- ☐ Vector averaging
- ☐ RMS averaging
- ☐ Peak hold

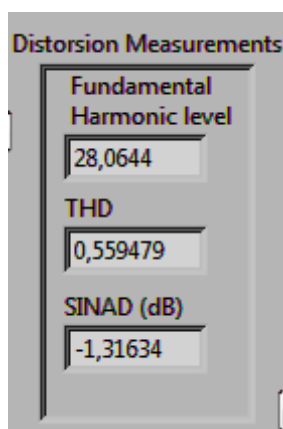
Se ofrecen los anteriores tipos de promediado, para visualizar mejor la respuesta en frecuencia y el enventanado, como ya se ha explicado en el capítulo de fundamentos teóricos.

### Weighting Mode:

- ☐ Linear
- ☒ Exponential

También se ofrecen los anteriores modos de ponderación.

### Mediciones de distorsión de señal:





### Códigos de error:

error in (no error)

status	code
<input checked="" type="checkbox"/>	0

source

Como en todas las aplicaciones y diseños comerciales, en sus inicios se esperan los reportes de fallos en situaciones no contempladas en el diseño. Para facilitar la depuración del mismo, poder actualizar la aplicación lo más rápido posible y ofrecer al cliente una solución rápida y fiable actualizando su aplicación, se proporciona un indicador de error, en el cual el usuario podrá reportar el código de error y la fuente del mismo. Esto facilitará en el futuro una respuesta rápida en el departamento de incidencias de la empresa en posesión de este proyecto.



Escuela Universitaria de Ingeniería  
Técnica de Telecomunicación  
Universidad Politécnica de Madrid





## **Bibliografía**

### **Capítulo planificación de recursos y costes**

- <http://www.gestiopolis.com/>
- Steve McConnell, "Desarrollo y gestión de proyectos informáticos". 1996
- Grady Booch, James Rumbaugh, Ivar Jacobson. "El lenguaje unificado de modelado". Ed.: Addison Wesley

### **Capítulo Ingeniería del Hardware:**

- Ramón Pallás Areny, " Adquisición y distribución de señales
- Manual de la tarjeta de adquisición de datos

### **Capítulo ingeniería del software**

- Documentos y manuales de usuario de Instrumentación virtual con Labview proporcionada por National Instruments.
- H. Goldberg, "What is virtual instrumentation?" IEEE Instrumentation & Measurement Magazine, vol. 3, no. 4, pp. 10–13, 2000.
- Prácticas de introducción a la Instrumentación Virtual. Programación en LabVIEW.
- Practicas de laboratorio de sistemas electrónicos de control cursada en mi titulación.

### **Capítulo de fundamentos teóricos**

#### Fundamentos de Análisis de Señales RF

- <http://zone.ni.com/wv/app/doc/p/id/wv-2129>
- <http://zone.ni.com/wv/app/doc/p/id/wv-2130>
- <http://zone.ni.com/wv/app/doc/p/id/wv-2131>

#### Guía para Medidas Más Comunes

- <http://zone.ni.com/devzone/cda/tut/p/id/7363>
- Digitizer/Oscilloscope Fundamentals



<http://zone.ni.com/devzone/cda/tut/p/id/3408>

¿Qué es Adquisición de Datos?

- <http://www.ni.com/dataacquisition/whatis/esa/>

Monitoreo y Registro Datos con Dataloggers

- <http://zone.ni.com/wv/app/doc/p/id/wv-625>

Implementación de Algoritmos de Procesamiento de Señales con LabVIEW y MathScript

- <http://zone.ni.com/wv/app/doc/p/id/wv-569>

Beneficios de Múltiples A/Ds en Aplicaciones de Adquisición de Datos

- <http://zone.ni.com/wv/app/doc/p/id/wv-642>

Tutorial sobre Comunicación Serial

- <http://zone.ni.com/devzone/cda/tut/p/id/5925>

Control de Instrumentos Autónomos y Modulares con NI LabVIEW SignalExpress

- <http://zone.ni.com/devzone/cda/tut/p/id/5798>

SignalExpress para la Enseñanza e Investigación

- <http://zone.ni.com/wv/app/doc/p/id/wv-419>

The Fundamentals of FFT-Based Signal Analysis and Measurement in LabVIEW and LabWindows/CVI

- <http://zone.ni.com/devzone/cda/tut/p/id/4278>

Legacy: Hardware Installation/Configuration Troubleshooter

- <http://www.ni.com/support/install/default.htm>

LabVIEW para Medición y Análisis de Datos

- <http://zone.ni.com/devzone/cda/tut/p/id/9613>

Beneficios de Múltiples A/Ds en Aplicaciones de Adquisición de Datos

- <http://zone.ni.com/wv/app/doc/p/id/wv-642>

Data Acquisition Reference Design for LabVIEW

- <http://zone.ni.com/devzone/cda/tut/p/id/11805>

Introduction to Storing Data with Data Acquisition



- <http://zone.ni.com/wv/app/doc/p/id/wv-1368>

Introduction to Using Data Acquisition for Monitoring Applications

- <http://zone.ni.com/wv/app/doc/p/id/wv-1366>

Codigo Labview Generador Funciones 2 & 4 canales

- <https://decibel.ni.com/content/docs/DOC-3545#comment-4060>

AWFG Arbitrary Waveform Function Generator

- <https://decibel.ni.com/content/docs/DOC-13739>

Busqueda seminarios y videos sobre generación de funciones con labview

- <http://search.ni.com/nisearch/app/main/p/bot/no/ap/tech/lang/es/pg/1/ps/30/sn/catnav:mm:ssnav:dzn/q/generator%20functions/>

## Capítulo de recursos adquiridos y mejoras

- Alexander, Christopher. (1977). A Pattern Language. Oxford University Press, Inglaterra.
- Boehm, B.W. (1981). Software Engineering Economics. Prentice-Hall, México.
- Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John. (1995). Design Patters Elements of Reusable Object-Oriented Software. Addison Wesley Publishing Company, Inc. USA.
- Sommerville, Ian. (2002). Ingeniería de Software. Addison Wesley Publishing Company, Inc. USA.